NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

PLANNING GERMAN ARMY HELICOPTER
MAINTENANCE
AND MISSION ASSIGNMENT

by

Achim Sgaslik

March, 1994

Thesis Advisor: Dr. Robert F. Dell

REPORT DOCUMENTATION PAGE

Form Approved OMB Np. 0704

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 1994	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE PLANNING GERMAN ARMY HELICOPTER MAINTENANCE AND MISSION ASSIGNMENT	5. FUNDING NUMBERS
6. AUTHOR(S) Sgaslik Achim	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/ MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13.

ABSTRACT *(maximum 200 words)*

German Army light helicopter transportation regiments operate 45 Bell UH-1D helicopters to support demanding missions throughout Europe. Maintenance period scheduling, major exercise and regular mission assignment decisions directly influence the readiness of the helicopter fleet. Currently, all planning is done manually, which is unstructured and time consuming. This thesis describes a decision support system designed to assist with maintenance planning and mission assignment. The yearly maintenance and event scheduling problem and the short term mission assignment tasks are formulated and solved as elastic mixed integer linear programs. Resulting yearly schedules and short term sortie plans are both generated in a fraction of the time previously required with solution quality superior to their manual counterparts.

14. SUBJECT TERMS Helicopter, Maintenance Planning, Mission Assignment.	15. NUMBER OF PAGES 107
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)

Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited.

Planning German Army Helicopter
Maintenance
and Mission Assignment

by

Achim Sgaslik
Captain, German Army
Dipl.Ing.(FH), University of Federal Armed Forces München, 1987

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1994

Author: ______________________
Achim Sgaslik

Approved by: ______________________
Robert F. Dell, Thesis Advisor

____
Gordon H. Bradley, Second Reader

____
Peter Purdue, Chairman
Department of Operations Research

ABSTRACT

German Army light helicopter transportation regiments operate 45 Bell UH-1D helicopters to support demanding missions throughout Europe. Maintenance period scheduling, major exercise and regular mission assignment decisions directly influence the readiness of the helicopter fleet. Currently, all planning is done manually, which is unstructured and time consuming. This thesis describes a decision support system designed to assist with maintenance planning and mission assignment. The yearly maintenance and event scheduling problem and the short term mission assignment tasks are formulated and solved as elastic mixed integer linear programs. Resulting yearly schedules and short term sortie plans are both generated in a fraction of the time previously required with solution quality superior to their manual counterparts.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This thesis develops two optimization models and proposes an interface for a self sufficient personal computer based decision support system as an interactive instrument to construct reliable and completely organized helicopter usage and maintenance plans. The optimization models assist with yearly maintenance and event scheduling, and short term helicopter - mission assignments. Computational experience shows both yearly schedules and short term sortie plans are generated in a fraction of the time previously required with solution quality superior to their manual counterparts.

These results are for a German Army light helicopter transportation regiment operating 45 Bell UH-1D helicopters. For such regiments, maintenance period scheduling, major exercise, and regular mission assignment decisions directly influence the readiness of the helicopter fleet. The planning supervisor in the regiment's maintenance and repair battalion strives to keep high technical and operational standards while meeting all necessary inspections (16 different inspections during a 1200 flight hours cycle), satisfying all mission and exercise requirements, equitably using the helicopters, and smoothly operating the maintenance facilities. Done manually, these tasks are unstructured and time consuming.

The two optimization models provide valuable assistance to the planning supervisor. The yearly planning model assigns

helicopters to inspections and to exercises while observing monthly planned flight hours and operational guidelines (a desired level of flight hour reserve, a percentage range of operationally ready helicopters, an upper level of monthly flight hours per helicopter, and inspection capacities). The short term model assigns helicopters to missions while observing the technical status of each helicopter (remaining flight hours to next inspection, planned time of next inspection, availability) and fulfilling all mission requirements with respect to flight hours, and equipment.

Resulting yearly schedules and computed short term mission assignment plans are face-valid (*i.e.* judged implementable by expert opinion), superior to their manual counterparts (*i.e.* fewer planning conflicts) and generated in a small fraction of the time previously required.

Included in this study are the system's structure, the development and implementation of the two optimization models, computational results, the interface proposal, and explanations regarding the planning process and intended user.

I. INTRODUCTION

German Army light transport helicopter regiments operate 45 Bell UH-1D helicopters (see Figure 1) in support of a corps with three army divisions. A maintenance officer or "*Leiter*

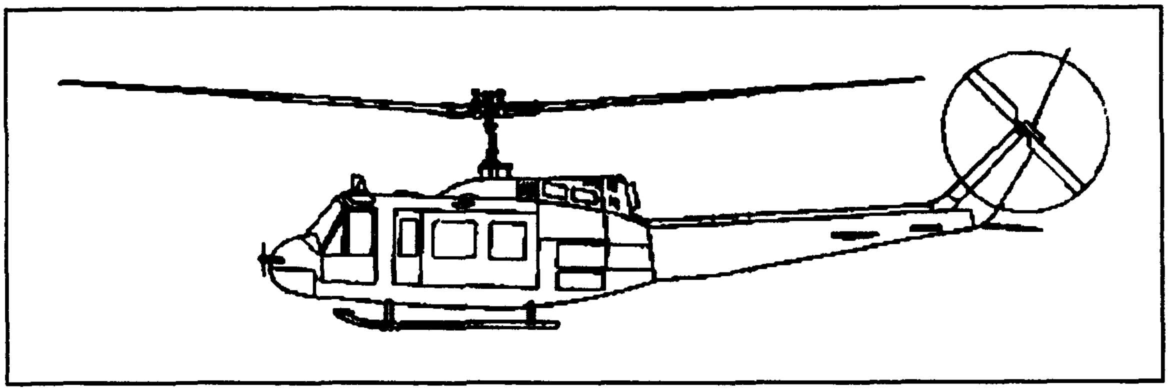

Figure 1

Einsatz" (the author's assignment from 1989 to 1991) in the regiment's maintenance and repair battalion supervises the helicopter fleet maintenance planning and mission assignment. He strives to keep high technical and operational standards while:

- Meeting all necessary inspections,
- Satisfying all mission requirements,
- Fulfilling special events such as NATO exercises,
- Equitably using the helicopters,
- Smoothly operating all maintenance facilities.

This thesis derives and solves integer linear programs to assist with maintenance planning and mission assignment.

A. BELL UH-1D INSPECTION SYSTEM

The Bell UH-1D maintenance cycle (in its German version) consists of 1200 flight hours and contains 16 different inspections in two levels (C and D). The C-level (see Table 1) includes relatively easy maintenance and part replacement, the D-level (see Table 2) contains costly repair overhauls lasting three to five weeks. Each inspection takes place 75 flight hours after its predecessor in the following order:

C1,C2,C1,D1,C1,C2,C1,D2,C1,C2,C1,D1a,C1,C2,C1,D3.

C and D inspections are accomplished in disjoint facilities with different technical personnel. The C inspections, being relatively easy, can be performed without using fixed installations (*e.g.* during exercises).

TABLE 1

C-LEVEL (Maintenance)	
NAME	DURATION
C1	1 to 3 working days
C2	2 to 5 working days

TABLE 2

D-LEVEL (Repair)		
NAME	DURATION	FLIGHT HOUR
D1	3 weeks	300
D2	4 weeks	600
D1a	4 weeks	900
D3	5 weeks	1200

B. CURRENT PLANNING PROCESS

The maintenance officer in charge of the planning and mission assignment tasks has a staff of up to six soldiers experienced in the fields of aircraft repair and maintenance. Despite access to a personal computer (PC), maintenance and mission assignments are currently based on poorly documented manual procedures which rely on complicated charts and overview boards.

The planning process breaks into two related pieces, yearly and short term. The yearly maintenance and event schedule takes a three man team up to five weeks to produce. The final product contains the flight hours assigned to each helicopter each month, helicopters assigned C and D inspections each month, and helicopters assigned to fly special events (*e.g.* operations of the Allied Mobile Force in northern Norway or in eastern Turkey, which require up to 12 helicopters to fly a total of more than 1000 hours). The short term plan provides individual helicopter mission assignments for up to one week. Time needed to develop the short term plan varies substantially depending on mission requirements and helicopter availability. A typical short term plan requires one man approximately two hours.

1. Yearly Maintenance and Event Schedule

Input to the yearly maintenance and event schedule includes:

- The total number of hours the helicopter fleet should fly,
- Special event requirements,
- Pilot Combat Training Programs (CTP, provide required monthly instructional flight hours),
- Operating data from past years on availability and reliability of the (aging) helicopter.

The yearly schedule has four primary operational guidelines:

- The flight hour reserve,
- The percentage of operationally ready helicopters,
- The upper level of monthly flight hours per helicopter,
- The maximum monthly C and D inspection capacities.

The flight hour reserve is each helicopter's available flight hours to the next D inspection, summed over all helicopters.

An ideal level of 6,750 hours (number of helicopters * 0.5 * hours between D inspections or 45 * 0.5 * 300 = 6,750) has historically been a good planning factor. A level above 7,500 hours has endangered future equitable use of the maintenance facilities (*i.e.* a disproportional number of helicopters require imminent inspections), a level below 5,500 hours has endangered the capability of the regiment to fulfill all required missions.

The percentage of operationally ready helicopters measures the number of helicopters not in a D inspection and with remaining hours to the next D inspections greater than zero. A level between 70% and 90% is desired.

No more than 30 flight hours should be assigned to each helicopter each month, but violations are sometimes necessary (*e.g.* for events).

The normal output capacity for D inspections is three per month. If planned well ahead, an output of four per month is achievable, but reserve capacity for exception repairs is lost and an equally high output in the succeeding month is unlikely. The C-inspection level capacity is easier to manage. Output variations from one to six helicopters in one week are possible.

2. Short Term Helicopter - Mission Assignments

The short term planning process attempts to pick the correct helicopter for each mission while observing:

- The number of flight hours each helicopter has until the next inspection
- The planned time and importance of the next inspection,
- The mission's flight hour requirement,
- The current equipment status and the mission's equipment requirement,
- The possible multiple use of the helicopter for non-simultaneous missions.

This daily task requires experience and talent. Unfortunately, an organized method of meeting the listed criteria is often blocked by time considerations.

C. OBJECTIVES OF THIS THESIS

The objective is to develop underlining algorithms and propose an interface for a self sufficient PC based decision support system as an interactive instrument to reach reliable and completely organized helicopter usage and maintenance plans.

The approach undertaken in this thesis is practical and centered around two optimization models.

Chapter II addresses similar concepts in the existing literature. Chapter III contains the developed algorithms, the proposed interface and both user and planning session descriptions for the decision support system. Chapter IV contains the computational performance of the developed models and Chapter V provides conclusions.

II. RELATED RESEARCH

A literature search did not reveal a model with the capabilities needed by the German Army Aviation maintenance officer. His tasks are very specific, dependent on the German version of the Bell UH-1D preventive maintenance system and the local maintenance resources. The combination of a yearly plan with a short term mission assignment system, the necessity of using this system independently and the exclusion of manpower, budget and logistic issues make the proposed model unique.

However, decision support and expert systems for aviation maintenance activities have been studied for other aircraft and organizations. Hackett and Pennartz (1982) provide the basis of a decision support system for the maintenance aircraft scheduling process of an United States (US) Air Force B-52 wing. They start with the principle that computerization without proper management does not improve overall performance. They therefore first establish maintenance responsibilities and management procedures for an operational cycle that considers everything from the yearly flight program to the daily scheduling update operations. They characterize maintenance scheduling as a complex process with a high degree of uncertainty, strict requirements, binding constraints and insufficient guidance. In their view, successful usage of optimization models in the different planning stages depends on appropriately reducing the complexity of the problem with simplifying and structuring techniques. They do not, however, develop optimization models or algorithms. Key aspects of a decision support system are reported as:

- Communicability with the user,
- Robustness even for extreme cases,
- Ease of Control.

Shenolikar (1983) describes a decision support system for automatic test equipment systems operations management, which is closely related to aircraft maintenance affairs. Optimization models are again addressed, but not formulated. The key elements of a generic decision support system are developed as follows:

- Knowledge base (with proposed algorithms and solution models),
- Data directory and data base,
- Report generator,
- Communications (interface) manager.

A series of theses (see Christensen and Pasadilla,1991) advised by Professor Martin J. McCaffrey (Department of Administrative Science, United States Naval Postgraduate School) develops a Naval Aviation Maintenance Organizational

Activity Strategic Information System (OASIS) and an Expert System Advisor for Aircraft Maintenance Scheduling (ESAAMS). These complex systems address flight and maintenance activities, but include also human resources (manpower management, training and qualification), monetary management (budget considerations and accounting) and inventory structures for logistical support. These programs deal with the combined use of present information systems of the US Naval Air Systems Command and design applicable knowledge bases, databases and graphical interfaces. Again, these systems only propose optimization models and algorithm as future work.

In addition to these expert systems for planning aircraft related maintenance and use, the literature search did reveal an optimization model concerned with helicopters and one dealing with military aircraft sortie planning. The "Phoenix" model (Brown, Clemence, Teufert and Wood, 1991) includes procurement and retirement schedules for the US Army's helicopter fleet, handling 16 different helicopter types over 25 years. The mission assignment problem is described by Wallace (1992), who develops sortie optimization tools for the US Air Force.

The search for related maintenance planning systems in use by commercial airlines was not successful. Commercial airlines are concerned with a different set of constraints (costs) and resources (see Talluri and Gopalan, 1993).

III. CONCEPT OF THE DECISION SUPPORT SYSTEM

A. STRUCTURE AND OVERVIEW

The system organization (see Figure 2) resembles the underlying manual planning process and consists of two almost independent parts:

- The yearly maintenance and event planning system,
- The short term planning system for mission assignment covering up to one week.

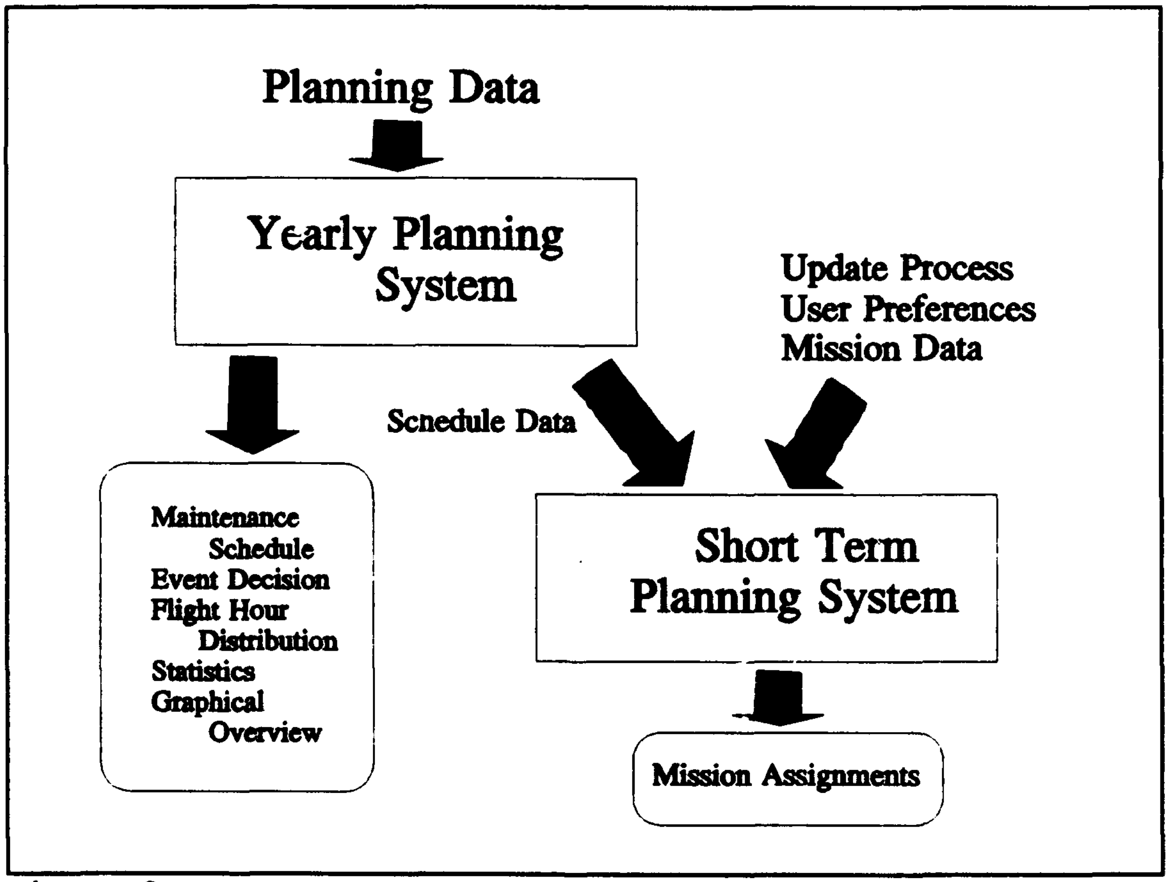

Figure 2

Each subsystem has the ability to respond to user preferences and demands. The scheduled inspection periods are input to the short term planning system which forms a connection between the two parts. There is no direct feedback out of the short term planning process into the yearly system, because a continuous change of the yearly plan destroys the underlying planning policy. There are good possibilities to get a yearly plan "back on track" when unforeseen trouble strikes. The experienced user has to decide, when a complete renewal of the yearly plan is necessary. Every subsystem result can and should be adapted manually. Results are proposals, providing feasible starting points for further planning and adapting to reality. All the important existing constraints are present in the different models, but elastic violations are possible (like real decision making) by paying adequate penalties.

B. THE YEARLY PLANNING SYSTEM

1. Basic Specifications

The yearly planning system forms the basis for all maintenance scheduling and mission assignment. It recognizes initial conditions and requirements from the planning data and implements the planning policy. The essence of this subsystem is an integer linear program. The necessary input can be expressed as:

- Planned flight hours per month for the fleet,

- Special event requirements (time, flight hours, number of helicopters, maintenance possibilities),
- D-level maintenance capacity,
- Status of each helicopter (remaining flight hours to next D inspection).

The final output includes :

- Recommended flight hours for each helicopter each month,
- Recommended D inspection decision for each helicopter each month,
- Helicopter assignment to special events,
- Monthly statistics with respect to flight hour supply and helicopter availability.

A Pascal program transforms the yearly solution (containing monthly decisions) into a solution containing weekly results and C inspection decisions. A description of this program follows the yearly planning model.

2. The Yearly Planning Model

The model specifications in a basic format can be described as follows:

■ <u>INDICES:</u>

t month (*e.g.* 1,2,...,12),

h helicopter identification (*e.g.* 1,2,...,45),

e event (*e.g.* 1,2,...,E).

■ <u>DATA:</u>

$PLHRS_t$ planned flight hours for month t,

$EVENTHRS_e$ flight hours required for each helicopter participating in event e,

$EVENTNMB_e$ — number of helicopters required for event e,

DMAX — maximum number of D inspections per month,

PRODHRS — additional flight hours (*e.g.* 300) obtained per inspection,

OPTSUP — desired flight hour supply (*e.g.* 6,750),

$MININSP_t$ — minimum number of inspections in month t,

MAXHRS — maximum regular monthly flight hours for each helicopter (*e.g.* 30).

■ VARIABLES:

BINARY VARIABLES

$Z_{h,e}$ — one if helicopter h is assigned to event e,

$Y_{h,t}$ — one if helicopter h is assigned a D inspection in month t.

POSITIVE VARIABLES

$X_{h,t}$ — flight hours assigned to helicopter h below MAXHRS in month t,

$XE_{h,t}$ — flight hours assigned to helicopter h in month t exceeding MAXHRS or in addition to an inspection or event assignment,

$REMHRS_{h,t}$ — flight hours until next D inspection for helicopter h at end of month t,

■ MODEL:

* Minimize the sum of all weighted elastic penalties and $XE_{h,t}$ hours.

* subject to the constraints:

$$(1) \qquad \sum_{h=1}^{45} [X_{h,t}+XE_{h,t}+\sum_{e=1}^{E} EVENTHRS_e * Z_{h,e}] \doteq PLHRS_t \qquad t=1..12$$

$$(2) \qquad \sum_{h=1}^{45} Y_{h,t} < \doteq DMAX \qquad t=1..12$$

$$(3) \qquad \sum_{h=1}^{45} Y_{h,t} \geq MININSP_t \qquad t=1..12$$

$$(4) \qquad REMHRS_{h,t} = REMHRS_{h,t-1} + PRODHRS * Y_{h,t} - X_{h,t} - XE_{h,t} - \sum_{e=1}^{E} EVENTHRS_e * Z_{h,e} \qquad h=1..45, t=1..12$$

$$(5) \qquad \sum_{h=1}^{45} REMHRS_{h,t} > \doteq OPTSUP \qquad t=1..12$$

$$(6) \qquad \sum_{t=1}^{12} [X_{h,t}+XE_{h,t}+\sum_{e=1}^{E} EVENTHRS_e * Z_{h,e}] \doteq (\sum_{t=1}^{12} PLHRS_t)/45 \qquad h=1..45$$

(7) $$\sum_{h=1}^{45} Z_{h,e} = EVENTNMB_e \qquad e=1..E$$

(8) $$Z_{h,e} + Y_{h,t} + (X_{h,t}/MAXHRS) \leq 1 \qquad h=1..45, e=1..E, t=1..12$$

■ <u>REMARKS:</u>

- The $\doteq, <\doteq, >\doteq$ signify elastic constraints.

- $REMHRS_{h,0}$ provide the initial status of each helicopter.

■ <u>DESCRIPTION OF THE CONSTRAINTS:</u>

(1) Meet planned flight hours each month or incur an elastic penalty.

(2) Comply with the maintenance capacity or incur an elastic penalty.

(3) Perform a minimum number of inspections each month.

(4) Calculate remaining hours until the next inspection for each helicopter at end of each month.

(5) Provide the desired flight hour supply or incur an elastic penalty.

(6) Provide equitable use of the helicopter fleet or incur an elastic penalty.

(7) Enforce number of required helicopters for each event.

(8) Allow each helicopter to participate only in a special event, be inspected, or be assigned regular flight hours $X_{h,t}$ without penalties each month. Assignment of flight hours $XE_{h,t}$ despite an event or inspection is possible, but incurs the penalty associated with the variable $XE_{h,t}$.

C. TRANSFORMATION OF OPTIMIZATION RESULTS

A PASCAL program named "MOSCH" (Monthly SCHedule) transforms the model solution into an editable form (see Appendix C). The objective is to translate the monthly decision variables into weekly results. Instead of operating with a general maintenance duration time for the D inspections of one month, the more realistic values for the respective D1,D2,D1a and D3 overhauls are utilized. The three required C inspections between every two D inspections are also added (recall the relative ease of planning these inspections). The basic approach can be described as follows:

- Process each helicopter ordered by initial flight hours until next D inspection.
- For each planned D-level inspection, schedule the inspection as soon as possible starting two weeks prior to the beginning of its planned inspection month. Ensure no planned hours or event conflict exists.
- If the starting week is feasible, check inspection workload and choose next available week with acceptable workload level.
- Schedule the inspection over its actual duration and check again for scheduling conflicts.
- Schedule all intermediate C inspections based on remaining hours to the next inspection for each helicopter.

- Note C inspection requirements during events.
- Print the schedule, a graphical overview and all required statistics.

D. THE SHORT TERM PLANNING SYSTEM

1. Basic Specification

The short term planning system assigns helicopters to missions while observing requirements of the yearly plan. It is proposed as an interactive program with a graphical user interface and an optimization model to perform mission assignment. Additional characterizations of the complete short term planning system are given later in this chapter.

The following addresses the optimization model, which can be used independently from the proposed system. Necessary input requirements for this mission assignment model are:

- Status of each helicopter with respect to availability, equipment, flight hours and inspection plan,
- Mission requirements with respect to flight hours and equipment.

The output includes an assignment proposal for each mission. Some sorties require a spare or backup helicopter to ensure mission success. The optimization model also decides the assignment of spare helicopters.

2. The Short Term Helicopter - Mission Assignment Model

This model chooses the best helicopter and (if required) spare helicopter for each mission. It takes the following helicopter properties into account:

- A user preference for mission assignment,
- The relative importance of the next inspection, expressed in weeks required for the completion,
- The time gap until the next planned inspection,
- The flight hours remaining until the next inspection,
- The current equipment condition,
- Restrictions for night or instrument flights,
- The overall availability during the planning time frame (no ongoing mission, event assignment or inspection).

Mission requirements are:

- Equipment,
- Flight hours,
- Backup demand,
- Time window,
- Night or instrument flight capability.

The equipment requirement is one of the following:

- 11 seats,
- 5 seats,
- 400 kg exercise load,
- Internal tank left,
- Internal tank right,
- Winch.

The number of items between any two equipment types on the above list (noncircular) represents the relative difficulty or time needed to change between the two equipment types. Therefore, a change between 400 kg exercise load and 5 seats is relative easy to do in comparison to a change between 11 seats and a winch. This approach is chosen, because data for a more realistic change-over matrix is currently not available.

A mission - helicopter assignment is only feasible, if the helicopter is available (expressed in a 0-1 availability subset), the helicopter is not assigned to a simultaneous mission, and mission requirements don't collide with flight restrictions of the helicopter. The mission requirements fall into one of the following codes:

- 0 = no operation limitations,
- 1 = helicopter restricted to Combat Training Flights (CTP) because of insufficient technical standard (*e.g.* non critical vibrations),
- 2 = helicopter restricted to Visual Flight Rules (VFR) missions because of absence of Instrumental Flight Rules (IFR) equipment,
- 3 = helicopter restricted to daylight flights only, because of absence of night sight equipment.

The model structure can be described as follows:

- <u>INDICES:</u>

h   helicopter identification (*e.g.* 1,2,...,45),

m   mission identification (*e.g.* 1,2,...,M),

i   simultaneous mission group (*e.g.* 1,2,...,I).

- <u>DATA:</u>

$Reminsp_h$ remaining hours to the next inspection for each helicopter h,

$Length_m$ flight hours for mission m,

$Costuse_{h,m}$	penalty for assigning helicopter h to mission m,
Spare	subset of all missions which require a spare helicopter,
$Overlap_i$	set of all missions in group i.

■ <u>VARIABLES:</u>

$FM_{h,m}$	binary assignment decision for helicopter h and mission m,
$FS_{h,m}$	binary assignment decision for helicopter h used as a spare for mission m.

■ <u>MODEL:</u>

* minimize

$$\sum_{h=1}^{45}\sum_{m=1}^{M} Costuse_{h,m} * (FM_{h,m}+FS_{h,m})$$

* subject to the constraints

$$(1) \qquad \sum_{h=1}^{45} FM_{h,m}=1 \qquad m=1..M$$

$$(2) \qquad \sum_{h=1}^{45} FS_{h,m}=1 \qquad m=1..M \;\; s.t. \;\; m \in Spare$$

(3) $\sum_{m=1}^{M}[Length_m*(FM_{h,m}+FS_{h,m})]\leq Reminsp_h$ $\quad h=1..45$

(4) $\sum_{m\in Overlap_i}[FM_{h,m}+FS_{h,m}]\leq 1$ $\quad h=1..45, i=1..I$

(5) $\sum_{m=1}^{M}FM_{h,m}<\doteq 1$ $\quad h=1..45$

■ <u>DESCRIPTION OF THE CONSTRAINTS:</u>

(1) Assign exactly one helicopter to mission m.

(2) Assign a spare helicopter to mission m if required.

(3) Each helicopter's flight hour use can not exceed its remaining hours until inspection.

(4) A helicopter can not be assigned to more than one mission being conducted at the same time.

(5) Assign each helicopter to at most one mission or incur an elastic penalty.

The variable $Costuse_{h,m}$ is a combination of weighted penalties for not choosing a helicopter with:

- the highest priority score,
- the most expensive following inspection (expressed in required completion time),

- the least expensive equipment changeover (expressed in equipment code differences),

- the smallest time gap to the following inspection.

Included is also a penalty for selecting a helicopter previously assigned a mission during the model's time frame.

A reformulation of the short term helicopter - mission assignment problem as a network flow model is possible. However, due to the success of the model described above (see computational results in Chapter IV) and the potential size of the network structure, this approach is not further investigated.

E. USER DESCRIPTION, PLANNING SESSIONS AND INTERFACE DESIGN

To fully understand the role of the two optimization models, it is necessary to present the context of the appropriate planning environment including a description of the users. The yearly model is primarily employed once a year by a well educated user and therefore can be used as demonstrated in chapter IV without an additional interface. The applicability of the short term model to assist with mission assignment is also demonstrated in chapter IV. Unfortunately, the intended user for the short term model lacks substantial familiarity with a personal computer. Therefore this section proposes a user interface for the short term model to assist with complex input and data manipulation needs.

1. Yearly Planning System

a. User Description

The yearly planning system user should have the following training, experience and educational requirements:

- Complete familiarity with the overall planning process,
- Computer skills including input/output problems and handling of textfiles,
- Basic skills in linear programming and familiarity with applicable software packages.

The use of the system by the maintenance officer in charge (usually with a master's degree in air and space technology or mechanical engineering) is recommended, but delegation to a computer experienced subordinate is possible. Every maintenance and repair battalion also has a "*S6-Offizier*", who is responsible for data processing and the computer facilities.

b. Description of a Planning Session

After preparing and screening the planning data (event information, the planned monthly flight hours, the maintenance facility capacities and the initial status of each helicopter), a textfile is filled with the required data in a simple coded form. Preferences and prefixed maintenance periods can be included. It is helpful to transform the achieved optimization model's solution into the more organized weekly schedule (now with C inspection recommendations included) for further analysis, using a transformation program as described above. Statistics and a graphical overview can be prepared for command reviews. The overall time frame for a planning session as described depends on the familiarity with the system, the solver, the capacity of the personal computer and the difficulty of the plan. Based on computational results (see Chapter IV), approximately two working days should be sufficient for multiple iterations of data input, computer runs, solution review, and manual modification of the schedule.

2. Short Term Planning System

In addition to the mission assignment optimization model as described above, the proposed system (see Figure 3) contains graphical interfaces including an update task screen for helicopter status changes, a report section that includes the maintenance status for each helicopter, and a screen for generating the updated short term maintenance plan.

a. *Interface Design*

Interface proposals for each screen can be found in Appendix E. These interfaces need a not yet developed communications manager for screen control and initiating computational intermediate steps in the knowledge base, using the database information.

b. *User Description*

The recommended user for this part of the planning system is one of the two senior NCO's in the technical command

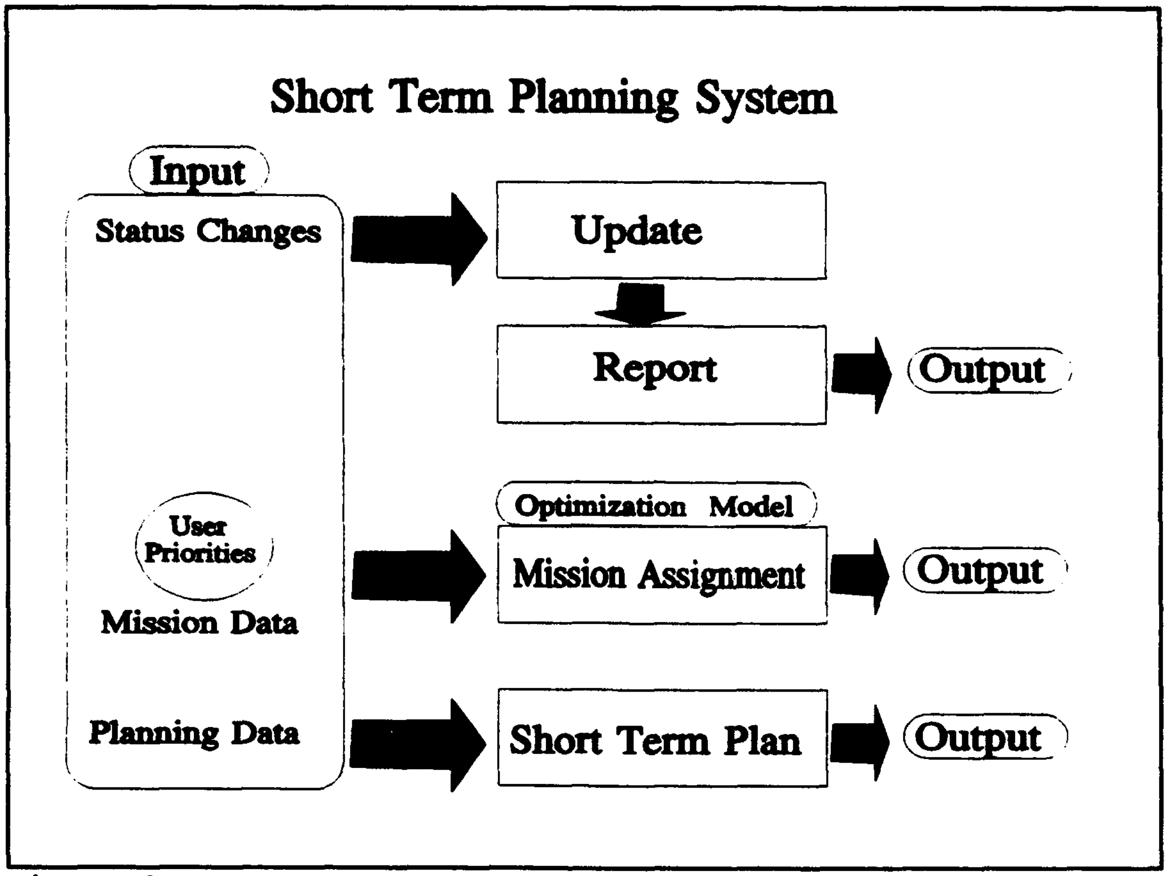

Figure 3

center. These are usually experienced soldiers with education as aircraft technicians and training in maintenance planning operations. Their computer skills can be basic. A practical briefing of a few days duration should be sufficient to enable this group of users to operate the subsystem. Expertise will be developed quickly during daily employment. The system design should include safety features against wrong input and confirmation procedures to prevent unintentional utilization. A simple interface structure, easy accessible, controllable and with convenient operation possibilities will ensure proper use of the system.

c. Description of a Planning Session

(1) Update Process. Every time the technical command center gets a vocal (telephone) or written report of helicopter status changes, an update screen allows the revision for the selected helicopter. These changes include:

- Failure of an operationally ready helicopter,
- Category of the failure,
- Estimated completion times for repairs or maintenance measures,
- Flight hour consumption,
- Equipment change overs,
- Special operation limitations,
- Renewed availability (mission or maintenance measurement completion).

The helicopter in question, identified by number, will be selected and a menu of instances appears. Each instance can be selected and changed within a range of offered possibilities. The confirmation of the changes concludes the update operation. An immediate report generation should be included.

(2) Mission Assignment Process. After determining the time frame for the missions in question (from one day to a week), a table asks for the mission requirements separated by a mission identification number. Simultaneous missions are marked with a common group index. Other needed data include flight hour requirements, estimated start and landing times,

description of the mission (combat training program or mission order number), spare helicopter requirement, equipment requirement and flight restrictions. The short term optimization model gives an assignment proposal for each mission and spare requirement. After reviewing the result, the user can accept the optimization proposal (in part or whole), rerun the optimization model, or manually edit the selection. The process finishes with a printed result (on screen or hardcopy) of the sortie plan.

(3) Reports and Statistics. A combined status report for a particular helicopter or the whole fleet can be selected at any time. This feature helps the supervisor in his control functions and generates required summaries for the next higher command levels. The statistics include:

- Percentage of available helicopters,
- Percentage of helicopters on mission,
- Percentage of helicopters in unplanned maintenance,
- Percentage of helicopters in planned maintenance,
- Flight hour reserve,
- Flight hour consumption so far for month and year,
- Available flight hours for rest of month and year.

Simple selection of the offered choices should be sufficient for generating the report and the statistics.

(4) Short Term Maintenance Plan. This special screen enables the user to issue his orders to the maintenance and repair level facilities. An updated extract of the yearly maintenance schedule is generated by selecting helicopters out of a candidate list, which includes helicopters already scheduled for inspection and those that will soon need inspection. The screen allows for comparison between the yearly plan and the updated short term version.

IV. COMPUTATIONAL EXPERIENCE

A. THE YEARLY MODEL

Two data sets (1989 and 1991) taken from the Army Aviation Regiment 30 in Niederstetten Air Base (Germany) show the applicability of the yearly model. The 1989 data (see Table 3) requires only two major events and is representative of a modestly difficult planning year. The 1991 data (see Table 4) pictures a difficult planning year consisting of four demanding events and a nonuniform yearly flight hour program. Unfortunately, the initial status of each helicopter (hours remaining until the next inspection and inspection type) is not known for either data set. Realistic estimates are based on the fleet's flight hour reserve, the percentage of operationally ready helicopters, and the author's personal experience. All computational results are obtained using an IBM compatible 486/33 Personal Computer. The model is generated using GAMS (Brooke, Kendrick and Meeraus, 1992) and solved using ZOOM (Brooke, Kendrick and Meeraus, 1992), XA (1993) and OSL (1991).

The GAMS code, which implements the yearly model can be found in Appendix A. The model reports approximately:

- 1,200 constraints in 10 blocks,
- 2,600 variables in 18 blocks (600 discrete and 2,000 continuous variables),

- 9,000 non zero elements.

Attempts to solve realistic instances of the yearly model optimally result in significant computational effort. Heuristic solution procedures were therefore investigated and found to provide quality solutions using substantially reduced

TABLE 3

DATA SET 1989

Month	Planned Flight Hours	Event Helicopter/Hours	
January	610		
February	630		
March	700		
April	610		
May	900	1[illegible]	80
June	727		
July	610		
August	680		
September	850		
October	680		
November	610		
December	749	5	65

TABLE 4

DATA SET 1991

Month	Planned Flight Hours	Event Helicopter/Hours	
January	526		
February	526		
March	1278	10	100
April	602		
May	1128	8	75
June	827		
July	526		
August	602		
September	1053	10	62
October	677		
November	537		
December	743	4	65

computation time. Two heuristic procedures are used, "LP-Rounding" and "Cascade". Both solution procedures rely on the relaxed integer solution being a valid information transfer tool for the pure integer task.

- LP-Rounding: After solving the linear programming relaxation, binary variables with values near zero or one are rounded respectively down or up to the nearest integer value and fixed. This process can be repeated indefinitely, but for this application two iterations diminish the number of decision variables sufficiently to allow for a successful solve of the reduced mixed integer program.

- Cascade: A number of possible implementations exist for this heuristic technique. For the test problems considered, solving twelve (one for each month) linear integer programs works best. The first of these twelve problems relaxes the integer restriction for variables associated with months two to twelve. After solving, a new problem is generated with month one's variables set equal to the values obtained in the previous relaxation, month two's variables constrained to be integer, and months three to twelve variables relaxed. This process repeats until the binary variables for all twelve months have been restricted to be integer.

Independent of the heuristic solution procedure, several parameter settings are especially important (see Appendix A for a complete list). After several test runs using various parameter values, best results are achieved for:

- An upper bound on $XE_{h,t}$ (flight hours assigned to helicopter h exceeding regular usage in month t) of 30,

- A penalty value of 0.1 for $XE_{h,t}$,
- A penalty value of 0.01 for the elastic variables of constraint (6), which provide for the equitable use of the helicopter fleet.

More important than the actual value is the relative magnitude of a penalty with respect to the other penalties. Specifically, inequable use of the helicopter fleet is viewed as an order of magnitude less important than excessive use of an individual helicopter (ratio 1/10).

Solutions from the different solvers (ZOOM, XA, OSL) and the two solution methods for the 1991 data set can be compared in Appendix B. ZOOM achieves its lowest costs schedules using a specified branching order based on the natural hierarchy of the time dependent discrete variables. The solvers XA and OSL are applied with the default branch and bound schemes.

The quality of the solutions does not depend on the solver or the heuristic, but ZOOM requires substantially more time than XA or OSL, and the Cascade procedure requires slightly more time than the LP-Rounding method. All achieved solutions exceed the initial relative optimality tolerance values (OPTCR) of 0.10 or 0.15 (guaranteeing a solution within 10% or 15% of optimal respectively). These tolerances apply to each individual program run and can therefore not serve as computation stopping conditions for the final solution. The solutions presented in Appendix B have objective function values between 11% and 36% of optimal (using an initial cost value of 100). However, the resulting schedules are face-valid (*i.e.* judged implementable by expert opinion) and considered superior to manually created plans.

Table 5 and 6 compare the manual and optimization model's yearly schedule for the two data sets. Without any additional manual editing, the selected system schedule for 1991 meets all requirements (percentage of operationally ready helicopters and flight hour reserve are in the desired ranges) and violates the realistic 30 hour planning goal and the planned hour per month constraints less than the manual solution (fewer planning conflicts). The computed schedule for the data set of 1989 shows similar advantages. The planned hours per month are now met exactly. Production time for both system schedules (including time needed for manual adjustment work) constitutes only a small proportion (approximately 1/12) of the usual manual completion time.

TABLE 5

COMPARISON SYSTEM SOLUTION - MANUAL ACHIEVED SOLUTIONS

Abbreviations: Plan = Planned Flight Hours (yearly flight program),

Man = Manual Planned Flight Hours

|Dev| = absolute Deviation from Pl. Hours, Com = Computed Planned Flight Hours

Plan 1991	Complicated set Solver: OSL Solution Procedure: Cascade												
Month	1	2	3	4	5	6	7	8	9	10	11	12	Total
Plan	526	526	1278	602	1128	827	526	602	1053	677	537	743	9025
Man	535	540	1250	600	1200	850	530	600	1000	700	540	750	9095
|Dev|	9	14	28	2	78	23	4	2	53	23	3	7	256
Com	526	526	1335	602	1184	827	526	604	1053	685	537	743	9148
|Dev|	0	0	57	0	56	0	0	2	0	8	0	0	123
Operationally ready helicopter percentage (70%..90% is desired)													Average
Man	82.2	82.2	75.6	77.8	75.6	82.2	82.2	80	80	82.2	84.4	82.2	80.55
Com	80	84.4	84.4	86.7	88.9	80	82.2	82.2	84.4	80	84.4	86.7	83.69
Deviation from 30 hours goal in hours													Total
Man	40	20	10	10	25	60	60	0	0	25	40	0	290
Com	0	0	35	0	25	14	5	20	0	30	0	17	146
Number of D-Inspection completions													Total
Man	4	2	2	3	2	4	2	2	3	3	3	2	32
Com	3	2	2	2	2	2	4	3	2	4	2	2	30
Flight hour reserve (desired level is 6750)													Average
Man	7171	7231	6581	6861	6381	6631	6701	6701	6601	6801	7161	7011	6813
Com	6800	6954	6219	6217	5632	5405	6079	6075	5622	6137	6208	6057	6123

TABLE 6

COMPARISON SYSTEM SOLUTION - MANUAL ACHIEVED SOLUTIONS

(for Abbreviations see Table 5)

Plan 1989	Simple set Solver: XA Solution procedure: LP-Rounding												
Month	1	2	3	4	5	6	7	8	9	10	11	12	Total
Plan	610	630	700	610	900	727	610	680	850	680	610	749	8356
Man	620	640	680	600	900	750	610	700	850	680	620	750	8400
|Dev|	10	10	20	10	0	23	0	20	0	0	10	1	104
Com	610	630	700	610	900	727	610	680	850	680	610	749	8356
|Dev|	0	0	0	0	0	0	0	0	0	0	0	0	0
Operationally ready helicopter percentage (70%..90% is desired)													Average
Man	82.2	84.4	80	77.8	75.6	80	77.8	75.6	71.1	68.9	68.9	68.9	75.93333
Com	82.2	86.7	82.2	80	77.8	77.8	77.8	75.6	68.9	71.1	71.1	71.1	76.85833
Deviation from 30 hours goal in hours													Total
Man	10	5	5	10	0	35	2	0	0	25	0	0	92
Com	0	35	20	0	0	2	10	10	0	0	0	0	77
Number of D-Inspection completions													Total
Man	3	2	2	2	2	3	2	2	3	3	2	2	28
Com	3	2	2	2	3	2	2	2	3	2	2	2	27
Flight hour reserve (desired level is 6750)													Average
Man	6786	6746	6666	6666	6366	6516	6506	6406	6456	6676	6656	6506	6579.33
Com	6796	6766	6666	6656	6656	6529	6519	6439	6409	6409	6399	6250	6547.833

B. THE SHORT TERM HELICOPTER - MISSION ASSIGNMENT MODEL

The short term model is tested using data based on past experience in mission assignment. A sample test problem (including solution) can be found in Tables 7A and 7B. This 20 mission example represents a normal one day assignment task. Initial input data is listed in Appendix D and XA is the solver. The model is also tested successfully with a highly complex 40 mission example. This case has demand for 34 missions with 18 spare requirements and for an unplanned exercise requesting six helicopters (five with 45 required flight hours each and one with 22 required flight hours).

The model handles the test cases smoothly when using an IBM compatible 486/33 Personal Computer. It produces solutions within one to three minutes. All achieved test solutions have objective values within five percent of optimality.

The GAMS code implementing the short term model can be found in Appendix D. The default branch and bound scheme for the XA solver is used.

A comparison of manual and computed assignments is difficult and not appropriate when acknowledging the very practical purpose of the system. A program of several months in an actual environment should be employed to evaluate the utility of the model. Tests with realistic mission sets however show consistent and appropriate helicopter selections, which pass the "common sense" test. Acknowledged advantages

TABLE 7A

MISSION ASSIGNMENT EXAMPLE

Mission	Required Equip.	Helo ID	Equip.	Priority	Next Insp.	Week of Insp
1	2	41	2	3	C2	15
2	2	3	3	3	D1	13
3	2	41	2	3	C2	15
4	3	33	3	3	C1	29
5	4	4	4	3	D2	13
6	5	42	5	3	C1	15
7	5	16	4	3	C1	21
8	3	40	3	2	D1	15
9	2	20	2	2	D3	25
10	1	41	2	3	C2	15
11	2	8	2	2	C1	15
12	5	18	6	3	D2	23
13	6	5	5	3	D3	14
14	1	9	3	2	C1	16
15	1	7	1	2	C1	15
16	1	39	1	1	D2	19
17	1	36	2	1	D2	27
18	5	17	5	3	D1	22
19	3	3	3	3	D1	13
20	2	20	2	2	D3	25

TABLE 7B

MISSION ASSIGNMENT EXAMPLE

Spare helicopter required for mission									
4	5	6	7	11	12	13	14	15	20
Selected spare helicopter									
32	16	17	17	14	12	17	20	14	8

of system solutions are the absence of conflicting assignments caused by user error (given correct input) and the substantially reduced development time.

V. CONCLUSION

This thesis develops two optimization models and proposes an interface for a self sufficient PC based decision support system as an interactive instrument to construct reliable and completely organized helicopter usage and maintenance plans.

The yearly maintenance and event planning model produces face-valid schedules in substantially less time than current manual techniques. The proposed procedure is structured, outcomes can be stored for reports, command reviews, and as a base for future planning.

Using the proposed interface to enhance data manipulation tasks, the short term planning model for mission assignment is able to produce daily to weekly sortie plans in minutes and free of user error.

Using integer linear programming as a foundation for a computer based maintenance scheduling and mission assignment system can substantially reduce the workload and improve the quality of this complex planning process.

APPENDIX A IMPLEMENTATION OF YEARLY PLANNING SYSTEM

A. LP-ROUNDING METHOD

```
      $TITLE   Helicopter Maintenance Scheduling
      $STITLE  A.Sgaslik  Thesis

* Computes a yearly schedule for 45 helicopters with respect
* to flight hour distribution, D inspections and events.

* The following program characteristics influence solutions drastically
* and need special attention:

* ● The relative termination tolerance OPTCR, which means that GAMS
*   will stop and report on the first solution found whose objective
*   value is within the specified tolerance of the best possible solution,
* ● The scalar DEVWEIGHT, which sets a penalty for no equitable use of the
*   helicopter fleet,
* ● The scalar EXPLWEIGHT, which sets a penalty for planning above the
*   monthly flight hour guideline for each helicopter,
* ● The restriction set of possible helicopters for event decisions R(T),
* ● The upper limits for flight hours for each helicopter and month X.UP
*   and XE.UP,
* ● The allowed percentage ranges for violating constraints (no penalty
*   violations).

*---------CONTROL OPTIONS--------------------------------
$OFFUPPER OFFSYMLIST OFFSYMXREF

OPTIONS
  LIMCOL = 0, LIMROW = 0, SOLPRINT = OFF ,DECIMALS = 2
  RESLIM = 500000, ITERLIM = 150000, OPTCR = 0.15, SEED = 3141
  OPTION LP = XA , RMIP = XA , MIP = XA ;

*---------DEFINITIONS AND DATA---------------------------
SETS
   T  month  / 0*12 /

   H  helicopter identification / 1*45 /

   E  event  / 1*4 /

   I  iterations for rounding method  / 1 /

ALIAS (T,TP);

PARAMETERS
   WEIGHT1(T)   penalty for deviations

   WEIGHT2(T)   penalty for deviations

   TOLERA(T)    tolerance for no penalty deviation of desired flight hour reserve

   U(T)  months without events
        / 1    1
          2    1
          3    0
          4    1
          5    0
          6    1
          7    1
          8    1
          9    0
          10   1
          11   1
          12   0 /


   PLHRS(T)   planned flight hours for month
          /1   526
           2   526
           3   1278
           4   602
```

```
          5   1128
          6   827
          7   526
          8   602
          9   1053
         10   677
         11   537
         12   743 /

INHRS(H)  initial remaining flight hours for each helicopter
              /45       0
               44       0
               43       0
               42       2
               41       5
               40       7
               1        10
               2        12
               3        13
               4        15
               5        20
               6        35
               7        45
               8        60
               9        75
               10       80
               11       90
               12       95
               13       110
               14       135
               15       145
               16       150
               17       155
               18       165
               19       175
               20       185
               39       195
               38       200
               37       205
               36       210
               35       220
               34       225
               33       230
               32       235
               31       240
               30       242
               29       250
               28       260
               27       270
               26       272
               25       278
               24       290
               23       300
               22       300
               21       300  /

 DINSP(H)  initial repair level inspection status for each helicopter
* code description   1=D1  2=D2  3=D1a  4=D3
              /1        1
               2        2
               3        3
               4        4
               5        1
               6        2
               7        3
               8        4
               9        1
               10       2
               11       3
               12       4
               13       1
               14       2
               15       3
               16       4
               17       1
               18       2
               19       3
               20       4
               21       1
               22       2
               23       3
               24       4
               25       1
               26       2
```

```
                  27        3
                  28        4
                  29        1
                  30        2
                  31        3
                  32        4
                  33        1
                  34        2
                  35        3
                  36        4
                  37        1
                  38        2
                  39        3
                  40        4
                  41        1
                  42        2
                  43        3
                  44        4
                  45        1     /

   EVENTHRS(E)  event requirements in hours for each helicopter
                 /1     100
                  2     75
                  3     62
                  4     65 /

   EVENTNMB(E) event requirements in helicopter numbers
                 /1       10
                  2       8
                  3       10
                  4       5 /

   INSPOK(H,T) one if helicopter - maintenance assignment possible

   EVENTOK(H,E,T) one if helicopter - event - time assignment possible;

TABLE
   S(T,E)   one when event in month

              1  2  3  4
          1
          2
          3   1
          4
          5      1
          6
          7
          8
          9         1
          10
          11
          12           1  ;

TABLE
   R(H,E)   one when helicopter event assignment initial possible

              1  2  3  4
         1    1  1     1
         2    1  1  1
         3    1  1  1  1
         4    1  1  1  1
         5       1     1
         6    1  1  1  1
         7    1  1  1  1
         8       1  1  1
         9    1     1  1
        10    1  1  1  1
        11    1  1  1  1
        12       1  1  1
        13    1  1     1
        14       1  1
        15    1  1  1  1
        16       1  1  1
        17
        18    1  1  1  1
        19    1  1     1
        20    1  1  1
        21    1  1  1  1
        22    1     1  1
        23    1  1  1  1
        24    1  1  1
```

```
  25   1  1  1  1
  26      1  1  1
  27   1  1     1
  29   1  1  1  1
  30   1  1     1
  31   1  1  1  1
  32
  33   1  1  1  1
  34   1  1  1  1
  35   1  1  1
  36      1  1  1
  37      1  1  1
  38   1  1  1  1
  39   1  1  1
  40      1  1  1
  41   1  1  1  1
  42   1  1  1  1
  43   1  1     1
  44      1  1  1
  45                  ;

SCALARS
   MDAYS       average maintenance days for one inspection
                    / 15 /

   DMAXDAYS    maximal repair level capacity in days per month
                    / 60 /

   PRODHRS     hour production per inspection
                    / 300 /

   OPTSUP      desired level of flight hour reserve

   DEVWEIGHT   penalty for not equitable fleet usage / 0.01 /

   EXPLWEIGHT  penalty for planning above 30 hours limit  / 0.1 /;

* parameter calculations

OPTSUP = CARD(H) * 150;
LOOP(T, WEIGHT1(T) = 0.5);
LOOP(T, WEIGHT2(T) = 1);

* Flexible tolerance for deviation from desired flight hour reserve
LOOP(T, IF (PLHRS(T) GT (4 * PRODHRS),      TOLERA(T) = 3 );
        IF ( (PLHRS(T) GE (2.5 * PRODHRS)) AND
             (PLHRS(T) LE (4 * PRODHRS)) ,TOLERA(T) = 2 );
        IF (  PLHRS(T) LT (2.5 * PRODHRS),TOLERA(T) = 1 ); );

INSPOK(H,T) = 1 $((((ORD(T)-1) * 100) GT INHRS(H)) AND
                    (ORD(T) GT 1)) ;

EVENTOK(H,E,T) = 1 $ ( (S(T,E) AND (R(H,E)) AND
  ( SUM(TP $ ((ORD(TP)) LE (ORD(T))), INSPOK(H,TP)) GT 0 ))
  OR ( S(T,E) AND (R(H,E)) AND
  ( SUM(TP $ ((ORD(TP)) LE (ORD(T))), INSPOK(H,TP)) EQ 0 ) AND
  (INHRS(H) - EVENTHRS(E) - ((((ORD(T)) - 1) * 100) GT 0) ) ) );

*-------------MODEL-------------------------------------

POSITIVE VARIABLES
   X(H,T)          assigned flight hours to helicopter in month

   XE(H,T)         assigned flight hours above 30 hour limit

   REMHRS(H,T)     flight hours until next inspection
                   for helicopter at end of month;

REMHRS.UP(H,T) = 300;


* Initialization of remaining hours until next inspection
LOOP(H,
       REMHRS.FX(H,'0') = INHRS(H));

* Upper limits for assigned flight hours in month
X.UP(H,T) = 30;
XE.UP(H,T) = 30;


BINARY VARIABLES
   Z(H,E)  one if helicopter is assigned to event e
```

```
   Y(H,T)  one if helicopter is assigned maintenance at month t;

* Fixing preplanned inspections in month 1
LOOP (H, IF (INHRS(H) EQ 0, Y.FX(H,'1') = 1 ;
                            X.FX(H,'1') = 0 ); );

POSITIVE VARIABLES

   DEVOPTSP1(T)   small penalty deviation from desired flight hour reserve

   DEVOPTSP2(T)   deviation from desired flight hour reserve

   DEVNOPEN1(T)   allowed deviation from desired flight hour reserve

   ELAST1(T)      elastic variable for equation plan

   ELAST2(T)      elastic variable for equation plan

   ELAST3(T)      elastic variable for equation maxd

   DEVFLEET1(H)   elastic variable for equitable fleet usage

   DEVFLEET2(H)   elastic variable for equitable fleet usage

   PLNOPEN(T)     allowed surplus planning hours in month

   PLSMPEN(T)     small penalty elastic variable for equation plan

   FLNOPEN1(H)    allowed deviation from equitable fleet usage

   FLNOPEN2(H)    allowed deviation from equitable fleet usage   ;

* Fixing no and small penalty limits
DEVOPTSP1.UP(T)   = 0.25 * OPTSUP;
DEVNOPEN1.UP(T)   = 0.10 * OPTSUP;
FLNOPEN1.UP(H)    = 0.35 * (SUM(T $ (ORD(T) GT 1), PLHRS(T)) / CARD(H));
FLNOPEN2.UP(H)    = 0.35 * (SUM(T $ (ORD(T) GT 1), PLHRS(T)) / CARD(H));
PLNOPEN.UP(T)     = 0.05 * PLHRS(T);
PLSMPEN.UP(T)     = 0.10 * PLHRS(T);

VARIABLE
   COST  objective function ;

EQUATIONS

   OBJ          objective function

   PLAN(T)      constraint on monthly planned hours

   MAXD(T)      repair level capacity constraint

   SMOD(T)      lower limit for inspections at month

   DEV1(T)      deviation from desired flight hour reserve constraint

   FLT1(H,T)    computation of remaining flight hours

   FLSM(H)      equitable fleet usage

   NUME(E,T)    exact number of helicopters for each event

   INS1(T,E,H)  no maintenance or assigned flight hours when event

   INS2(T,H)    no assigned hours when maintenance ;

* minimize

OBJ..   COST =E= 100 + SUM(T $ (ORD(T) GT 1),
       ( WEIGHT1(T) * (PLSMPEN(T) + DEVOPTSP1(T)) ) +
       ( WEIGHT2(T) * ( DEVOPTSP2(T) + ELAST1(T)
       + ELAST2(T) + ELAST3(T)) ) )
       + DEVWEIGHT  * SUM(H,DEVFLEET1(H) + DEVFLEET2(H))
       + EXPLWEIGHT * SUM( (T,H) $ (ORD(T) GT 1), XE(H,T) ) ;

* subject to

PLAN(T) $ (ORD(T) GT 1)..
     SUM(H , X(H,T) +  XE(H,T) +
     SUM(E $ (S(T,E)),EVENTHRS(E) * Z(H,E) $ (EVENTOK(H,E,T)))) =E=
     PLHRS(T) - ELAST2(T) + ELAST1(T) + PLNOPEN(T) + PLSMPEN(T);
```

```
MAXD(T) $ (ORD(T) GT 1)..  SUM(H ,MDAYS * Y(H,T) $ INSPOK(H,T))
                           =L= DMAXDAYS + ELAST3(T) ;

SMOD(T) $ (ORD(T) GT 1)..  SUM (H, Y(H,T) $ INSPOK(H,T)) =G= 2;

DEV1(T) $ (ORD(T) GT 1)..
     SUM(H, REMHRS(H,T)) =G= OPTSUP - DEVOPTSP1(T) - DEVOPTSP2(T) -
     TOLERA(T) * DEVNOPEN1(T)  ;

FLT1(H,T) $ (ORD(T) GT 1)..
     REMHRS(H,T) =E= REMHRS(H,T-1) - X(H,T) - XE(H,T) -
     SUM(E $ (S(T,E)) ,EVENTHRS(E) * (Z(H,E) $ (EVENTOK(H,E,T)))) +
     PRODHRS * (Y(H,T) $ INSPOK(H,T)) ;

FLSM(H)..     SUM(T $ (ORD(T) GT 1), X(H,T) + XE(H,T) +
              SUM( E ,EVENTHRS(E) * (Z(H,E) $ (EVENTOK(H,E,T))))) =E=
              (SUM( T $ (ORD(T) GT 1), PLHRS(T)) / CARD(H) ) +
              DEVFLEET1(H) - DEVFLEET2(H) + FLNOPEN1(H) - FLNOPEN2(H) ;

NUME(E,T) $ (S(T,E))..
              SUM(H, Z(H,E) $ (EVENTOK(H,E,T)) ) =E= EVENTNMB(E) ;

INS1(T,E,H) $ (S(T,E))..   Z(H,E) $ (EVENTOK(H,E,T)) +  X(H,T) / 30 +
                           (Y(H,T) $ INSPOK(H,T)) =L= 1;

INS2(T,H) $ (U(T))..   X(H,T) / 30 + (Y(H,T) $ INSPOK(H,T)) =L= 1;


MODEL  HELICOPTER   /ALL/;
SOLVE  HELICOPTER   USING  RMIP  MINIMIZING COST;


* iterative solving

LOOP(I,
   LOOP(H,
      LOOP(E, IF (Z.L(H,E) GT 0.95 , Z.FX(H,E) = 1 );
              IF (Z.L(H,E) LT 0.05 , Z.FX(H,E) = 0 );
         LOOP(T $ S(T,E), IF (Z.L(H,E) EQ 1 , Y.FX(H,T) = 0 ;
                                              X.FX(H,T) = 0 ); ); );
      LOOP(T, IF (Y.L(H,T) LT 0.05 , Y.FX(H,T) = 0 );
         LOOP(E $ S(T,E), IF (Y.L(H,T) EQ 1 , Z.FX(H,E) = 0 ;
                                              X.FX(H,T) = 0 ); ); ); );

            SOLVE  HELICOPTER   USING  RMIP  MINIMIZING COST;   );


SOLVE  HELICOPTER   USING  MIP  MINIMIZING COST;


*---------REPORTS-------------------------------------

PARAMETERS REPORT1(*,T)  planned hours for each helo and month;
   REPORT1 (H,T)   = X.L(H,T) + XE.L(H,T) + SUM(E $ S(T,E),
                     EVENTHRS(E) * Z.L(H,E)) ;
   REPORT1 ('TOTAL',T) = SUM (H, X.L(H,T) + XE.L(H,T) +
                         SUM(E $ S(T,E), EVENTHRS(E) * Z.L(H,E)));
   REPORT1 ('PLAN ',T) = PLHRS(T);

PARAMETERS REPORT2(*,T)  maintenance decision for helo and month ;
   REPORT2 (H,T) $ ( ORD(T) GT 1 )  = Y.L(H,T);
   REPORT2 ('TOTAL',T) $ ( ORD(T) GT 1 ) = SUM( H, Y.L(H,T));

PARAMETERS REPORT3(*,E)  event decision for helo and event;
   REPORT3 (H,E)   = Z.L(H,E);
   REPORT3 ('TOTAL',E) = SUM(H, Z.L(H,E));

PARAMETERS REPORT4(T)  summed maintenance day for each month;
   REPORT4 (T) $ (ORD(T) GT 1) =  SUM( H, MDAYS * Y.L(H,T));

PARAMETERS REPORT5(*,T)  flight hour reserve for each helo and month;
   REPORT5 (H,T) = REMHRS.L(H,T);
   REPORT5 ('TOTAL',T) = SUM (H, REMHRS.L(H,T));
   REPORT5 ('OPTIM',T) = 6750;

PARAMETERS REPORT6(T) percentage of ready to fly helicopters;
   REPORT6 (T) $ (ORD(T) GT 1) = 100 -
           (100 * SUM (H $ ((REMHRS.L(H,T) EQ 0) OR
           (Y.L(H,T) EQ 1)),1) / CARD(H));

OPTION REPORT1:2:1:1;
DISPLAY REPORT1;
```

```
OPTION REPORT2:2:1:1;
DISPLAY REPORT2;

OPTION REPORT3:2:1:1;
DISPLAY REPORT3;

DISPLAY REPORT4;

OPTION REPORT5:2:1:1;
DISPLAY REPORT5;

DISPLAY REPORT6;

DISPLAY PLSMPEN.L,ELAST1.L,ELAST2.L;
DISPLAY ELAST3.L;
DISPLAY DEVOPTSP1.L,DEVOPTSP2.L;
DISPLAY DEVFLEET1.L,DEVFLEET2.L;
DISPLAY XE.L;


* Creating input for Pascal transformation program Mosch
FILE RES /INPUTFIL.PAS/ ;
PUT RES;

PUT "INPUTFILE MODEL HELICOPTER" ;
PUT /;
PUT /;
LOOP (H,
       LOOP (T $(ORD(T) GT 1), PUT REPORT1(H,T):5:0 );
       PUT / );
PUT /;
LOOP (H,
       LOOP (T $(ORD(T) GT 1), PUT Y.L(H,T):3:0 );
       PUT / );
PUT /;
PUT CARD(E):2:0 /;
LOOP (H,
       LOOP (E, PUT Z.L(H,E):3:0 );
       PUT / );
PUT /;
LOOP (H, PUT DINSP(H):3:0 ;
         PUT /  );
PUT /;
PUT /;
LOOP (H,
       LOOP (T, PUT REMHRS.L(H,T):4:0 );
       PUT / );
PUT /;
```

B. CASCADE METHOD

Remark: Only parts different to A. are listed.

```
SETS
   T  month  / 0*12 /

   H  helicopter identification / 1*45 /

   E  event  / 1*4 /

   I  iterations  / 1*12 /


VARIABLE
   COST  objective function ;


EQUATIONS

   OBJ           objective function

   PLANA(T)      constraint on monthly planned hours
   PLANB(T)      constraint on monthly planned hours
   MAXDA(T)      repair level capacity
   MAXDB(T)      repair level capacity

   SMODA(T)      lower limit for inspections at month
```

```
   SMODB(T)       lower limit for inspections at month

   DEV1(T)        deviation of desired flight hour reserve constraint

   FLT1A(H,T)     computation of remaining flight hours
   FLT1B(H,T)     computation of remaining flight hours

   FLSM(H)        equitable fleet usage

   NUMEA(E,T)     exact number of helicopters for each event
   NUMEB(E,T)     exact number of helicopters for each event

   INS1A(T,E,H)   no maintenance and planned hours when event
   INS1B(T,E,H)   no maintenance and planned hours when event

   INS2A(T,H)     no planned hours when maintenance
   INS2B(T,H)     no planned hours when maintenance ;

* minimize

OBJ..   COST =E= 100 + SUM(T $ (ORD(T) GT 1),
        ( WEIGHT1(T) * (PLSMPEN(T) + DEVOPTSP1(T)) ) +
        ( WEIGHT2(T) * ( DEVOPTSP2(T) + ELAST1(T)
        + ELAST2(T) + ELAST3(T)) ) )
        + DEVWEIGHT  * SUM(H,DEVFLEET1(H) + DEVFLEET2(H))
        + EXPLWEIGHT * SUM( (T,H) $ (ORD(T) GT 1), XE(H,T) ) ;


* subject to

PLANA(T) $ ((ORD(T) GT 1) AND (ORD(T) LE A))..
    SUM(H , X(H,T) +  XE(H,T) +
    SUM(E $ (S(T,E)),EVENTHRS(E) * Z(H,E) $ (EVENTOK(H,E,T)))) =E=
    PLHRS(T) - ELAST2(T) + ELAST1(T) + PLNOPEN(T) + PLSMPEN(T);

PLANB(T) $ (ORD(T) GT A)..
    SUM(H , X(H,T) +  XE(H,T) +
    SUM(E $ (S(T,E)),EVENTHRS(E) * ZC(H,E) $ (EVENTOK(H,E,T)))) =E=
    PLHRS(T) - ELAST2(T) + ELAST1(T) + PLNOPEN(T) + PLSMPEN(T);

MAXDA(T) $ ((ORD(T) GT 1) AND (ORD(T) LE A))..
           SUM(H ,MDAYS * Y(H,T) $ INSPOK(H,T))
           =L= DMAXDAYS + ELAST3(T) ;
MAXDB(T) $ (ORD(T) GT A)..
           SUM(H ,MDAYS * YC(H,T) $ INSPOK(H,T))
           =L= DMAXDAYS + ELAST3(T) ;

SMODA(T) $ ((ORD(T) GT 1) AND (ORD(T) LE A))..
           SUM (H, Y(H,T) $ INSPOK(H,T)) =G= 2;
SMODB(T) $ (ORD(T) GT A)..
           SUM (H, YC(H,T) $ INSPOK(H,T)) =G= 2;

DEV1(T) $ (ORD(T) GT 1)..
    SUM(H, REMHRS(H,T)) =G= OPTSUP - DEVOPTSP1(T) - DEVOPTSP2(T) -
    TOLERA(T) * DEVNOPEN1(T)  ;

FLT1A(H,T) $ ((ORD(T) GT 1) AND (ORD(T) LE A))..
    REMHRS(H,T) =E= REMHRS(H,T-1) -
    X(H,T) - XE(H,T) -
    SUM(E $ (S(T,E)) ,EVENTHRS(E) * (Z(H,E) $ (EVENTOK(H,E,T)))) +
    PRODHRS * (Y(H,T) $ INSPOK(H,T)) ;
FLT1B(H,T) $ (ORD(T) GT A)..
    REMHRS(H,T) =E= REMHRS(H,T-1) -
    X(H,T) - XE(H,T) -
    SUM(E $ (S(T,E)) ,EVENTHRS(E) * (ZC(H,E) $ (EVENTOK(H,E,T)))) +
    PRODHRS * (YC(H,T) $ INSPOK(H,T)) ;

FLSM(H).. SUM(T $ ((ORD(T) GT 1) AND (ORD(T) LE A)), X(H,T)   XE(H,T) +
   SUM( E ,EVENTHRS(E) * (Z(H,E) $ (EVENTOK(H,E,T))))) +
   SUM(T $ (ORD(T) GT A), X(H,T) + XE(H,T) +
   SUM( E ,EVENTHRS(E) * (ZC(H,E) $ (EVENTOK(H,E,T))))) =E=
   (SUM( T $ (ORD(T) GT 1), PLHRS(T)) / CARD(H) ) + DEVFLEET1(H)
   - DEVFLEET2(H) + FLNOPEN1(H) - FLNOPEN2(H) ;

NUMEA(E,T) $ (S(T,E) AND (ORD(T) LE A))..
   SUM(H, Z(H,E) $ (EVENTOK(H,E,T)) ) =E= EVENTNMB(E) ;
NUMEB(E,T) $ (S(T,E) AND (ORD(T) GT A))..
   SUM(H, ZC(H,E) $ (EVENTOK(H,E,T)) ) =E= EVENTNMB(E) ;

INS1A(T,E,H) $ (S(T,E) AND (ORD(T) LE A))..
   Z(H,E) $ (EVENTOK(H,E,T)) +  X(H,T) / 30 + (Y(H,T) $ INSPOK(H,T)) =L= 1;
INS1B(T,E,H) $ (S(T,E) AND (ORD(T) GT A))..
   ZC(H,E) $ (EVENTOK(H,E,T)) +  X(H,T) / 30 + (YC(H,T) $ INSPOK(H,T)) =L= 1;

INS2A(T,H) $ (U(T) AND (ORD(T) LE A))..
```

```
      X(H,T) / 30 + (Y(H,T) $ INSPOK(H,T)) =L= 1;
INS2B(T,H) $ (U(T) AND (ORD(T) GT A))..
      X(H,T) / 30 + (YC(H,T) $ INSPOK(H,T)) =L= 1;

MODEL  HELICOPTER   /ALL/;

* iterative solving

LOOP (I, A = ORD(I) + 1 ;
      SOLVE  HELICOPTER   USING  MIP  MINIMIZING COST ;
      LOOP (H,
            LOOP (T $ (ORD(T) LE A), X.FX(H,T) = X.L(H,T) ;
                                     XE.FX(H,T) = XE.L(H,T);
                                     Y.FX(H,T) = Y.L(H,T) ;
                                     REMHRS.FX(H,T) = REMHRS.L(H,T) ;
                  LOOP(E $ S(T,E),   Z.FX(H,E) = Z.L(H,E) ); ); ); );
```

APPENDIX B TABULATED SOLUTIONS OF THE YEARLY SCHEDULE

Computer:		IBM comp. 486/33		
Sol.procedure: Solver:		LP-Rounding ZOOM		
OPTCR	Obj.Val.	Approximate time frame:		
0.15	128.53	3 hours		
Month	D-Insp.	Reserve	%Avail.	Σ XE
1	3	6853	86.7	20
2	2	6901	82.2	30
3	2	6159	82.2	0
4	3	6427	80.0	0
5	4	6499	80.0	0
6	2	6272	77.8	28
7	2	6346	77.8	30
8	4	6944	82.2	30
9	3	6783	82.2	30
10	3	7006	84.4	30
11	3	7366	88.9	30
12	4	7823	88.9	30

Total Number of D-level inspections : 35

Total XE hours : 258

Computer:		IBM comp. 486/33		
Sol.procedure: Solver:		Cascade ZOOM		
OPTCR	Obj.Val.	Approximate time frame:		
0.15	111.07	4 hours		
Month	D-Insp.	Reserve	%Avail.	Σ XE
1	3	6880	84.4	0
2	2	6954	84.4	0
3	4	6876	82.2	0
4	4	7474	86.7	0
5	2	6946	88.9	0
6	4	7303	86.7	30
7	2	7377	91.1	0
8	4	7975	91.1	29
9	2	7522	91.1	35
10	2	7445	93.3	0
11	2	7508	93.3	5
12	2	7328	95.6	7

Total Number of D-level inspections : 33

Total XE hours : 106

Computer:		IBM comp. 486/33		
Sol.procedure: Solver:		LP-Rounding XA		
OPTCR	Obj.Val.	Approximate time frame:		
0.1	115.86	15 minutes		
Month	D-Insp.	Reserve	%Avail.	Σ XE
1	4	7180	73.3	47
2	2	7254	77.8	30
3	2	6576	82.2	3
4	2	6574	75.6	52
5	3	6346	75.6	0
6	2	6119	77.8	0
7	2	6193	77.8	0
8	2	6191	80.0	0
9	3	6083	71.1	0
10	3	6227	66.7	0
11	2	6290	66.7	0
12	2	6147	64.4	0

Total Number of D-level inspections : 29

Total XE hours : 132

Computer:		IBM comp. 486/33		
Sol.procedure: Solver:		Cascade XA		
OPTCR	Obj.Val.	Approximate time frame:		
0.10	136.29	20 minutes		
Month	D-Insp.	Reserve	%Avail.	Σ XE
1	3	6853	86.7	7
2	2	6901	84.4	7
3	4	6739	86.7	20
4	2	6757	86.7	60
5	2	6229	84.4	60
6	2	6002	77.8	0
7	2	6075	77.8	0
8	2	6073	80.0	0
9	2	5567	80.0	0
10	4	6075	77.8	56
11	3	6438	75.6	62
12	2	6295	80.0	30

Total Number of D-level inspections : 30

Total XE hours : 242

Computer:		IBM comp. 486/33		
Sol.procedure: Solver:		LP-Rounding OSL		
OPTCR	Obj.Val.	Approximate time frame:		
0.10	133.35	15 minutes		
Month	D-Insp.	Reserve	%Avail.	Σ XE
1	4	7161	82.2	100
2	2	7209	91.1	37
3	2	6538	88.9	90
4	3	6811	88.9	60
5	3	6583	86.7	30
6	2	6536	86.7	0
7	2	6430	86.7	0
8	3	6700	86.7	0
9	2	6235	86.7	0
10	2	6158	82.2	0
11	2	6218	82.2	0
12	2	6075	82.2	0

Total Number of D-level inspections : 29

Total XE hours : 317

Computer:		IBM comp. 486/33		
Sol.procedure: Solver:		Cascade OSL		
OPTCR	Obj.Val.	Approximate time frame:		
0.10	124.40	20 minutes		
Month	D-Insp.	Reserve	%Avail.	Σ XE
1	3	6880	80.0	0
2	2	6954	84.4	0
3	2	6219	84.4	35
4	2	6217	86.7	0
5	2	5632	88.9	25
6	2	5405	80.0	14
7	4	6079	82.2	5
8	2	6075	82.2	20
9	2	5622	84.4	0
10	4	6137	80.0	30
11	2	6200	84.4	0
12	2	6057	86.7	17

Total Number of D-level inspections : 29

Total XE hours : 146

APPENDIX C TRANSFORMATION PROGRAM FOR YEARLY SCHEDULE

A. PASCAL CODE

Compiler: Borland Pascal Version 6.0 (1990)

```
program Mosch (input,output);
{$r+}

uses Crt;

{    Author      : Achim Sgaslik
     Assignment  : Thesis "Planning German Army Helicopter
                   Maintenance and Mission Assignment"

     Written     : 10/10/93
     Update      :
     Objective   : translate GAMS generated monthly helicopter
                   inspection and event planning into a weekly
                   schedule                                      }

{------ constant and type definition part ------------------------}

const MAX = 45;
{ number of helicopters }


type Idtype = 1..MAX;
{ identification numbers for the helicopters }

type Dtype = 1..4;
{ D inspections identification D1,D2,D1a,D3}

type Inspidtype = 1..16;
{ complete inspection cycle, both levels }

type Hrstype = integer;
{ flight hours }

type Monthtype = 1..12;

type Binartype = 0..1;

type Insparraytype = array[Idtype] of Dtype;

type Hrsarraytype = array[Idtype] of Hrstype;

type Montharraytype = array[Monthtype] of integer;

type Matrixarraytype = array[Idtype,0..12] of Hrstype;

type Bimatrixtype = array[Idtype,Monthtype] of Binartype;


{ for initial input translation and later use as update structure }
type Structype = record
                    Idfield     : Idtype;
                    Reminsp     : Hrstype;
                    Nextinsp    : Inspidtype;
                 end;

type Inputstructype = array[Idtype] of Structype;


{ schedule subtypes }
type Dschedtype = record
                     Dfield  : Dtype;
                     Idnr    : Idtype;
                     Occup   : boolean;
                  end;
```

```
type Cschedtype = record
                  Idnr      : Idtype;
                  Occup     : boolean;
                 end;

type Eventtype = record
                  Nr          : integer;
                  Idnr        : Idtype;
                  Occup       : boolean;
                  Inspinevent : integer;
                end;


{ Schedule  main types }
type DScheduletype = record
                      Dschedarray    : array[1..6] of Dschedtype;
                      WorkD          : integer;
                     end;

type DISchedulestructype = array[1..50] of DScheduletype;


type CScheduletype = record
                      Cschedarray    : array[1..6] of Cschedtype;
                      WorkC          : integer;
                     end;

type CISchedulestructype = array[1..50] of CScheduletype;


type EvScheduletype = record
                       Eventarray   : array[1..15] of Eventtype;
                      end;

type EventSchedulestructype = array[1..50] of EvScheduletype;



{ for the Z event decision variable from opt. model}
type Inevarraytype = array[Idtype,1..10] of Binartype;


{------ variable declaration part -------------------------------}

var Dinsp   : Insparraytype;         { initial inspection status D - insp }
    Remhrs  : Matrixarraytype;       { Remhrs [H,T] }
    Inithrs : Hrsarraytype;          { Inithrs [H]  }
    Y       : Bimatrixtype;          { Y [H,T] inspection decision }
    X       : Matrixarraytype;       { X [H,T] + XE [H,T] planned hours }
    SumY    : Montharraytype;        { Sum (T, Y [H,T]   }
    DSched  : DISchedulestructype;
    C1Sched : CISchedulestructype;
    C2Sched : CISchedulestructype;
    EvSched : EventSchedulestructype;
    Input   : Inputstructype;
    Z       : Inevarraytype;         { Z [H,E]  event decision  }
    Start   : -2..48;                { Starting week for scheduling D - insp }
    Abwch   : integer;               { Deviation variable }
    Inputfile,Eventfile,Outputfile    : text;
    K,H,T,KO,KM,O,KB,XM,N,L,NO        : integer;
    Finished,Eventinway,Eventmarker   : boolean;


{------ procedure and function declaration part ------------------}

procedure Sort ( var SInput : Inputstructype ; SRemhrs : Matrixarraytype ;
                 var SInithrs : Hrsarraytype );

{ Sorting of inputstructure in order of remhrs }

var Buffer1,Buffer2 : Structype;
    Buffer3,Buffer4 : Hrstype;
    J,Q             : integer;
    Smallest        : integer;

begin
   for J := 1 to (MAX-1) do begin
      Smallest := J;
      for Q := (J + 1) to MAX do begin
         if SInithrs[Q] < SInithrs[Smallest] then begin
            Smallest := Q;
         end;
```

```
      end;

      if Smallest > J then begin
         Buffer1 := SInput[Smallest];
         Buffer2 := SInput[J] ;
         Buffer3 := SInithrs[Smallest];
         Buffer4 := SInithrs[J];
         SInput[Smallest] := Buffer2;
         SInput[J] := Buffer1;
         SInithrs[Smallest] := Buffer4;
         SInithrs[J] := Buffer3;
      end;
   end;
end; {Sort}


procedure Formtranslate ( FoRemhrs : Matrixarraytype ;
                  var FInput : Inputstructype ; FDinsp   : Insparraytype ;
                  var FInithrs : Hrsarraytype );

{ translates initial data and fills inputstructure }

var H : integer;

begin
   for H := 1 to MAX do begin
      with FInput[H] do begin
         Idfield := H;
         FInithrs[H] := FoRemhrs[H,0];

         case FDinsp[H] of
            1 : begin
                  if FoRemhrs[H,0] <=75 then begin
                    Reminsp  := FoRemhrs[H,0];
                    Nextinsp := 4;
                  end;
                  if (FoRemhrs[H,0] > 75) and (FoRemhrs[H,0] <= 150)
                   then begin
                    Reminsp  := FoRemhrs[H,0] - 75;
                    Nextinsp := 3;
                  end;
                  if (FoRemhrs[H,0] > 150) and (FoRemhrs[H,0] <= 225)
                   then begin
                    Reminsp  := FoRemhrs[H,0] - 150;
                    Nextinsp := 2;
                  end;
                  if (FoRemhrs[H,0] > 225)  then begin
                    Reminsp  := FoRemhrs[H,0] - 225;
                    Nextinsp := 1;
                  end;
                end;
            2 : begin
                  if (FoRemhrs[H,0] <= 75) then begin
                    Reminsp  := FoRemhrs[H,0];
                    Nextinsp := 8;
                  end;
                  if (FoRemhrs[H,0] > 75) and (FoRemhrs[H,0] <= 150)
                   then begin
                    Reminsp  := FoRemhrs[H,0] - 75;
                    Nextinsp := 7;
                  end;
                  if (FoRemhrs[H,0] > 150) and (FoRemhrs[H,0] <= 225)
                   then begin
                    Reminsp  := FoRemhrs[H,0] - 150;
                    Nextinsp := 6;
                  end;
                  if FoRemhrs[H,0] >= 226 then begin
                    Reminsp  := FoRemhrs[H,0] - 225;
                    Nextinsp := 5;
                  end;
                end;
            3 : begin
                  if (FoRemhrs[H,0] <= 75) then begin
                    Reminsp  := FoRemhrs[H,0];
                    Nextinsp := 12;
                  end;
                  if (FoRemhrs[H,0] > 75) and (FoRemhrs[H,0] <= 150)
                   then begin
                    Reminsp  := FoRemhrs[H,0] - 75;
                    Nextinsp := 11;
                  end;
                  if (FoRemhrs[H,0] > 150) and (FoRemhrs[H,0] <= 225)
                   then begin
                    Reminsp  := FoRemhrs[H,0] - 150;
```

```
                    Nextinsp := 10;
                  end;
                  if FoRemhrs[H,0] >= 226 then begin
                    Reminsp  := FoRemhrs[H,0] - 225;
                    Nextinsp := 9;
                  end;
                end;
           4 : begin
                  if (FoRemhrs[H,0] <= 75) then begin
                    Reminsp  := FoRemhrs[H,0];
                    Nextinsp := 16;
                  end;
                  if (FoRemhrs[H,0] > 75) and (FoRemhrs[H,0] <= 150)
                   then begin
                    Reminsp  := FoRemhrs[H,0] - 75;
                    Nextinsp := 15;
                  end;
                  if (FoRemhrs[H,0] > 150) and (FoRemhrs[H,0] <= 225)
                   then begin
                    Reminsp  := FoRemhrs[H,0] - 150;
                    Nextinsp := 14;
                  end;
                  if (FoRemhrs[H,0] > 225) then begin
                    Reminsp  := FoRemhrs[H,0] - 225;
                    Nextinsp := 13;
                  end;
                end;
         end;
      end;
   end;
end; {Formtranslate}


function GetInspLength ( GDinspfield : Dtype) : integer;

begin
   case GDinspfield of
      1 : GetInspLength := 3;
      2 : GetInspLength := 4;
      3 : GetInspLength := 3;
      4 : GetInspLength := 5;
   end;
end; {GetInspLength}


procedure Initialize (var IDSched : DISchedulestructype ;
                      var IC1Sched,IC2Sched : CISchedulestructype ;
                      var IEvSched : EventSchedulestructype);

var  M,I1,I2,I3,I4 : integer;

begin
   for M := 1 to 50 do begin
      with IDSched[M] do begin
         WorkD := 0;
         for I1 := 1 to 6 do begin
            Dschedarray[I1].Occup := False;
         end;
      end;
      with IC1Sched[M] do begin
         WorkC := 0;
         for I2 := 1 to 6 do begin
            CSchedarray[I2].Occup := False;
         end;
      end;
      with IC2Sched[M] do begin
         WorkC := 0;
         for I3 := 1 to 6 do begin
            CSchedarray[I3].Occup := False;
         end;
      end;
      with IEvSched[M] do begin
         for I4 := 1 to 15 do begin
            Eventarray[I4].Nr := 0;
            Eventarray[I4].Occup := False;
            Eventarray[I4].Inspinevent := 0;
         end;
      end;
   end;
end; {Initialize}
```

```
procedure Exfill (var EX     : Matrixarraytype      ;
                  var EEvSched : Eventschedulestructype ;
                  var EY    : Bimatrixtype    ; var ESumY : Montharraytype;
                  var EZ : Inevarraytype      ; var EDinsp : Insparraytype;
                  var ERemhrs : Matrixarraytype;
                  var EInputfile,EEventfile : text );

{ fills all variables with results from optimization model from inputfile }

var H,E,T,I,L,K,Buffer,Nrofevents    : integer;
    Insp                             : Binartype;
    Eventweek,Eventlength            : integer;

begin
   readln (EInputfile);
   readln (EInputfile);
   for H := 1 to 45 do begin
      for T := 1 to 12 do begin
         read (EInputfile,EX[H,T]);
      end;
      readln (EInputfile);
   end;
   readln (EInputfile);

   for H := 1 to MAX do begin
      for T := 1 to 12 do begin
         read (EInputfile,EY[H,T]);
      end;
      readln (EInputfile);
   end;
   readln (EInputfile);

   for T := 1 to 12 do begin
      Buffer := 0;
      for H := 1 to MAX do begin
         if EY[H,T] = 1 then begin
            Buffer := Buffer + 1;
         end;
      end;
      ESumY[T] := Buffer;
   end;

   readln (EInputfile,Nrofevents);
   for H := 1 to MAX do begin
      for E := 1 to Nrofevents do begin
         read (EInputfile,EZ[H,E]);
      end;
      readln (EInputfile);
   end;

   readln (EInputfile);
   for H := 1 to MAX do begin
      readln (EInputfile,EDinsp[H]);
   end;

   readln (EInputfile);
   for H := 1 to MAX do begin
      for T := 0 to 12 do begin
         read (EInputfile,ERemhrs[H,T]);
      end;
      readln (EInputfile);
   end;

   for H := 1 to MAX do begin
      for E := 1 to Nrofevents do begin

         if EZ[H,E] = 1 then begin
            for I := 1 to E do begin
               readln (EEventfile);
            end;
            read (EEventfile,Eventweek);
            read (EEventfile,Eventlength);
            reset (EEventfile);

            for L := 0 to (Eventlength -1) do begin
               K := 1;
               while EEvSched[Eventweek + L].Eventarray[K].Occup do begin
                  K := K + 1;
               end;
               EEvSched[Eventweek + L].Eventarray[K].Idnr := H;
               EEvSched[Eventweek + L].Eventarray[K].Occup := True;
               EEvSched[Eventweek + L].Eventarray[K].Nr := E;
            end;
         end;
      end;
```

```
   end;
end; {Exfill}


procedure Printstatistic (var GOutputfile : text ; GRemhrs : Matrixarraytype;
                          GY : Bimatrixtype );

var Buffer,GT,GH : integer;

begin
   writeln (GOutputfile);
   writeln (GOutputfile);
   writeln (GOutputfile,'Percentage of operationally ready Helicopters');
   for GT := 1 to 12 do begin
      Buffer := 0;
      for GH := 1 to MAX do begin
         if (GRemhrs[GH,GT] = 0) or (GY[GH,GT] = 1) then begin
            Buffer := Buffer + 1;
         end;
      end;
      writeln (GOutputfile,
        (100 - ((Buffer/MAX) * 100)):3:1,' %  at Month ',GT:3);
   end;
   writeln (GOutputfile);
end; {Printstatistic}


procedure Printhours (var TOutputfile : text ; TX : Matrixarraytype ;
                      TSumY : Montharraytype );

var GT,GH : integer;

begin
   writeln (TOutputfile);
   writeln (TOutputfile,' Planned Hours per month and helicopter');
   write (TOutputfile,'          ');
   for GT := 1 to 12 do begin
     write (TOutputfile,GT:3,' ');
   end;
   writeln (TOutputfile);
   writeln (TOutputfile);
   for GH := 1 to MAX do begin
      write (TOutputfile,GH:3,'      ');
      for GT := 1 to 12 do begin
         write (TOutputfile,TX[GH,GT]:3,' ');
      end;
      writeln (TOutputfile);
   end;
   writeln (TOutputfile);
   writeln (TOutputfile);
   writeln (Toutputfile,' Number of D - inspections per month ');
   for GT := 1 to 12 do begin
      writeln (TOutputfile,TSumY[GT]:3,'  inspections in month ',GT);
   end;
   writeln (TOutputfile);
   writeln (TOutputfile);
end; { Printhours }


function ScheduleC ( SKM : integer ) : integer;

{ computes start week for C - inspections w.r.t  remaining hours
  to the inspections }

begin
   if SKM < 10 then begin
      ScheduleC := 0;
   end;
   if SKM in [10..20] then begin
      ScheduleC := 1;
   end;
   if SKM in [21..30] then begin
      ScheduleC := 2;
   end;
   if SKM >= 31 then begin
      ScheduleC := 3;
   end;
end; {ScheduleC}
```

```
procedure Printinputstruc (PInput : Inputstructype ; var POutputfile : text;
                           PRemhrs : Matrixarraytype );

var H : integer;

begin
   writeln (POutputfile,' Helicopterdata ');
   writeln (POutputfile,' Nr   ID   Remhrs  Nextinsp  init.Remhrs');
   for H := 1 to MAX do begin
      with PInput[H] do begin
         write (POutputfile,H:3,'  ');
         write (POutputfile,Idfield:3,'  ');
         write (POutputfile,Reminsp:6,'  ');
         write (POutputfile,Nextinsp:7,'  ');
         writeln (POutputfile,PRemhrs[PInput[H].Idfield,0]:7);
      end;
   end;
   writeln (POutputfile);
end; {Printinputstruc}


procedure Printschedule (PEvSched : EventSchedulestructype ;
                          PDSched : DISchedulestructype ;
                PC1Sched,PC2Sched : CISchedulestructype ;
                  var POutputfile : text );

var M,K : integer;

begin
   writeln (POutputfile);
   writeln (POutputfile);
   writeln (POutputfile,'************ ');
   writeln (POutputfile,'* Schedule * ');
   writeln (POutputfile,'************ ');
   writeln (POutputfile);
   writeln (POutputfile,' Code for Inspections during an event : ');
   writeln (POutputfile,' 1 = C1 prior to a C2');
   writeln (POutputfile,' 2 = C2                ');
   writeln (POutputfile,' 3 = C1 after a C2   ');
   writeln (POutputfile,'*********************************************');
   for M := 1 to 50 do begin
      writeln (POutputfile,'Week =  ',M);
      writeln (POutputfile,'D-Schedule');
      K := 1;
      while PDSched[M].Dschedarray[K].Occup do begin
         write (POutputfile,'ID = ',PDSched[M].Dschedarray[K].Idnr:3,'  ');
         writeln (POutputfile,
           ' with D-Inspection : ',PDSched[M].Dschedarray[K].Dfield,'  ');
         K := K + 1;
      end;

      writeln (POutputfile);
      writeln (POutputfile,'C1-Schedule');
      K := 1;
      while PC1Sched[M].Cschedarray[K].Occup do begin
         writeln (POutputfile,
           'ID = ',PC1Sched[M].Cschedarray[K].Idnr:3,'  ');
         K := K + 1;
      end;

      writeln (POutputfile);
      writeln (POutputfile,'C2-Schedule');
      K := 1;
      while PC2Sched[M].Cschedarray[K].Occup do begin
         writeln (POutputfile,
           'ID = ',PC2Sched[M].Cschedarray[K].Idnr:3,'  ');
         K := K + 1;
      end;

      writeln (POutputfile);
      writeln (POutputfile,'Event-Schedule');
      K := 1;
      while PEvSched[M].Eventarray[K].Occup do begin
         write (POutputfile,
           'Eventnumber = ',PEvSched[M].Eventarray[K].Nr,'  ');
         write (POutputfile,
           'ID = ',PEvSched[M].Eventarray[K].Idnr:3,'  ');
         writeln (POutputfile,
           'Inspection = ',PEvSched[M].Eventarray[K].Inspinevent);
         K := K + 1;
      end;
      writeln (POutputfile);
      writeln (POutputfile,'*********************************************');
```

```
   end;
end; {Printschedule}


procedure Eventcheck (var VEvSched : EventSchedulestructype ;
                      VT : Monthtype ; var VEventmarker : boolean ;
                      Ind : integer );

{ checks if and which inspection is due during event month }

var KMM,LM : integer;

begin
   VEventmarker := False;
   for KMM := 0 to 3 do begin
      LM := 1;
      if VEvSched[(T-1)*4+1+KMM].Eventarray[LM].Nr > 0 then begin
         while VEvSched[(VT-1)*4+1+KMM].Eventarray[LM].Occup do begin
            if VEvSched[(VT-1)*4+1+KMM].Eventarray[LM].Idnr = H then begin
               VEvSched[(VT-1)*4+1+KMM].Eventarray[LM].Inspinevent := Ind;
               VEventmarker := True;
            end;
            LM := LM + 1;
         end;
      end;
   end;
end; {Eventcheck}



procedure Graphicschedule ( var IDSched : DISchedulestructype ;
                            var IC1Sched,IC2Sched : CISchedulestructype ;
                            var IEvSched : EventSchedulestructype ;
                            var IOutputfile : text );

var LM,IM,IH : integer;
    Fieldwritten : boolean;

begin
   writeln (IOutputfile);
   writeln (IOutputfile,' Graphical Schedule ');
   writeln (IOutputfile);
   writeln (IOutputfile,' 1 = D1   2 = D2   3 = D1a   4 = D3 '),
   writeln (IOutputfile,' * = C1   $ = C2   E = Event        ');
   writeln (IOutputfile,' Week = quarter of a month on horizontal axis');
   writeln (IOutputfile,' Helicopter ID on vertical axis     ');
   writeln (IOutputfile);
   write (IOutputfile,'    ');
   for IM := 1 to 16 do begin
      write (IOutputfile,IM:2,' ');
   end;
   writeln (IOutputfile);
   writeln (IOutputfile);
   for IH := 1 to MAX do begin
      write (IOutputfile,IH:2,' ');
      for IM := 1 to 16 do begin
         Fieldwritten := False;
         LM := 1;
         if IEvSched[IM].Eventarray[LM].Nr > 0 then begin
            while (IEvSched[IM].Eventarray[LM].Occup) do begin
               if IEvSched[IM].Eventarray[LM].Idnr = IH then begin
                  write (IOutputfile,' E ');
                  Fieldwritten := True;
               end;
               LM := LM + 1;
            end;
         end;

         LM := 1;
         while IDSched[IM].Dschedarray[LM].Occup do begin
            if IDSched[IM].Dschedarray[LM].Idnr = IH then begin
               write (IOutputfile,' ');
               write (IOutputfile,IDSched[IM].Dschedarray[LM].Dfield:1,' ');
               Fieldwritten := True;
            end;
            LM := LM + 1;
         end;

         LM := 1;
         while IC1Sched[IM].Cschedarray[LM].Occup do begin
            if IC1Sched[IM].Cschedarray[LM].Idnr = IH then begin
               write (IOutputfile,' * ');
               Fieldwritten := True;
```

```
         end;
         LM := LM + 1;
      end;

      LM := 1;
      while IC2Sched[IM].Cschedarray[LM].Occup do begin
         if IC2Sched[IM].Cschedarray[LM].Idnr = IH then begin
            write (IOutputfile,' $ ');
            Fieldwritten := True;
         end;
         LM := LM + 1;
      end;

      if not Fieldwritten then begin
         write (IOutputfile,' - ') ;
      end;
   end;
   writeln (IOutputfile);
end;
writeln (IOutputfile);
writeln (IOutputfile);
write (IOutputfile,'    ');
for IM := 17 to 32 do begin
   write (IOutputfile,IM:2,' ');
end;
writeln (IOutputfile);
for IH := 1 to MAX do begin
   write (IOutputfile,IH:2,' ');
   for IM := 17 to 32 do begin
      Fieldwritten := False;
      LM := 1;
      if IEvSched[IM].Eventarray[LM].Nr > 0 then begin
         while (IEvSched[IM].Eventarray[LM].Occup) do begin
            if IEvSched[IM].Eventarray[LM].Idnr = IH then begin
               write (IOutputfile,' E ');
               Fieldwritten := True;
            end;
            LM := LM + 1;
         end;
      end;

      LM := 1;
      while IDSched[IM].Dschedarray[LM].Occup do begin
         if IDSched[IM].Dschedarray[LM].Idnr = IH then begin
            write (IOutputfile,' ');
            write (IOutputfile,IDSched[IM].Dschedarray[LM].Dfield:1,' ');
            Fieldwritten := True;
         end;
         LM := LM + 1;
      end;

      LM := 1;
      while IC1Sched[IM].Cschedarray[LM].Occup do begin
         if IC1Sched[IM].Cschedarray[LM].Idnr = IH then begin
            write (IOutputfile,' * ');
            Fieldwritten := True;
         end;
         LM := LM + 1;
      end;

      LM := 1;
      while IC2Sched[IM].Cscheda      M].Occup do begin
         if IC2Sched[IM].Sched        ].Idnr = IH then begin
            write (IOutputfile,
            Fieldwritten := True
         end;
         LM := LM + 1;
      end;

      if not Fieldwritten then begin
         write (IOutputfile,' - ') ;
      end;
   end;
   writeln (IOutputfile);
end;

writeln (IOutputfile);
writeln (IOutputfile);
write (IOutputfile,'    ');
for IM := 33 to 48 do begin
   write (IOutputfile,IM:2,' ');
end;
writeln (IOutputfile);
for IH := 1 to MAX do begin
   write (IOutputfile,IH:2,' ');
```

```
      for IM := 33 to 48 do begin
         Fieldwritten := False;
         LM := 1;
         if IEvSched[IM].Eventarray[LM].Nr > 0 then begin
            while (IEvSched[IM].Eventarray[LM].Occup) do begin
               if IEvSched[IM].Eventarray[LM].Idnr = IH then begin
                  write (IOutputfile,' E ');
                  Fieldwritten := True;
               end;
               LM := LM + 1;
            end;
         end;

         LM := 1;
         while IDSched[IM].Dschedarray[LM].Occup do begin
            if IDSched[IM].Dschedarray[LM].Idnr = IH then begin
               write (IOutputfile,' ');
               write (IOutputfile,IDSched[IM].Dschedarray[LM].Dfield:1,' ');
               Fieldwritten := True;
            end;
            LM := LM + 1;
         end;

         LM := 1;
         while IC1Sched[IM].Cschedarray[LM].Occup do begin
            if IC1Sched[IM].Cschedarray[LM].Idnr = IH then begin
               write (IOutputfile,' * ');
               Fieldwritten := True;
            end;
            LM := LM + 1;
         end;

         LM := 1;
         while IC2Sched[IM].Cschedarray[LM].Occup do begin
            if IC2Sched[IM].Cschedarray[LM].Idnr = IH then begin
               write (IOutputfile,' $ ');
               Fieldwritten := True;
            end;
            LM := LM + 1;
         end;

         if not Fieldwritten then begin
            write (IOutputfile,' - ') ;
         end;
      end;
      writeln (IOutputfile);
   end;
end; {Graphicschedule}


{------ main program statement part ------------------------------}

begin
   assign (Eventfile,'C:\TP\STUDY\Eventfile.pas');
   assign (Inputfile,'C:\TP\STUDY\Inputfile.pas');
   assign (Outputfile,'C:\TP\STUDY\Outputfile.pas');

   rewrite (Outputfile);
   reset (Inputfile);
   reset (Eventfile);


   Initialize (DSched,C1Sched,C2Sched,EvSched);
   Exfill (X,EvSched,Y,SumY,Z,Dinsp,Remhrs,Inputfile,Eventfile);
   Formtranslate (Remhrs,Input,Dinsp,Inithrs);
   Sort (Input,Remhrs,Inithrs);

   for K := 1 to MAX do begin
      H := Input[K].Idfield;
      for T := 1 to 12 do begin

         { Scheduling D - inspections }
         if Y[H,T] = 1 then begin
            N := GetInspLength (Dinsp[H]);
            if T > 1 then begin
               { initial starting point for scheduling D - inspection
                 second week in month prior to required completion }
               Start := (T-1)*4 - 1;
               Abwch := 0;

               for O := 0 to 1 do begin
                  { check if event prohibits prior scheduling }
                  L := 1;
                  if EvSched[Start + O].Eventarray[L].Nr > 0 then begin
```

```
    while EvSched[Start + 0].Eventarray[L].Occup do begin
       if EvSched[Start + 0].Eventarray[L].Idnr = H
        then begin
          Abwch := Abwch + 1;
        end;
       L := L + 1;
    end;
  end;
end;

{ no advance scheduling if planned hours
  in previous month > 30 }
if ((Abwch = 0) or (Abwch = 1)) and (X[H,T-1] > 30) then begin
  Abwch := 2;
end;

end
{ special case first month }
else begin
   Start := -1;
   Abwch :=  2;
end;

Finished := False;
while not Finished do begin
   { if workload in week too big or if planned hours in
     required completion month, search for next
     available spot }
   if (DSched[Abwch + Start].WorkD > 2) or (X[H,T] > 0)
     then begin
       case (N + Abwch) of
           3,4,5 : begin
                     Abwch := Abwch + 1;
                   end;
           6     : begin
                     { check if event prohibits inspection period
                       reaching into next month  }
                     Eventinway := False;
                     L := 1;
                     if EvSched[N + Start + Abwch].
                          Eventarray[L].Nr > 0 then begin
                        while (EvSched[N + Start + Abwch].
                           Eventarray[L].Occup)
                             and (L <= 5) do begin
                           if (EvSched[N + Start + Abwch].
                             Eventarray[L].Idnr = H) then begin
                              Eventinway := True;
                           end;
                           L := L + 1;
                        end;
                     end;

                     if not Eventinway then begin
                        Abwch := Abwch + 1;
                     end
                     else begin
                        Finished := True;
                     end;

                   end;
           7     : begin

                     Eventinway := False;
                     L := 1;
                     if EvSched[N + Start + Abwch].
                        Eventarray[L].Nr > 0 then begin
                         while (EvSched[N + Start + Abwch].
                           Eventarray[L].Occup)
                             and (L <= 5) do begin
                             if (EvSched[N + Start + Abwch].
                               Eventarray[L].Idnr = H) then begin
                                Eventinway := True;
                             end;
                             L := L + 1;
                         end;
                     end;

                     if not Eventinway then begin
                        Abwch := Abwch + 1;
                     end;
                     Finished := True;

                   end;
       end;
```

```
            end
            else begin
               Finished := True;
            end;
         end;

         { record inspection in schedule and update workload }
         for NO := 0 to (N-1) do begin

            KO := 1;
            while (DSched[Start + Abwch + NO].Dschedarray[KO].Occup)
              and (KO <= 5) do begin
               KO := KO + 1;
               if KO >= 5 then begin
                  writeln (Outputfile,
' Schedule conflict D inspection for helicopter ',H,' in month ',T);
               end;
            end;

            DSched[Start + Abwch + NO].Dschedarray[KO].Occup := True;
            DSched[Start + Abwch + NO].Dschedarray[KO].Dfield := Dinsp[H];
            DSched[Start + Abwch + NO].Dschedarray[KO].Idnr := H;
            DSched[Start + Abwch + NO].WorkD :=
              DSched[Start + Abwch + NO].WorkD + 1;
         end;

         { update initial inspection variable for the case of a
           second inspection during planning period }
         if Dinsp[H] = 4 then begin
            Dinsp[H] := 1
         end
         else begin
            Dinsp[H] := (Dinsp[H]) + 1;
         end;
      end;


      { Scheduling C1,C2 - Inspections using remaining hours in month
        and to next inspection }
      if (Remhrs[H,T] <= 75) and (Remhrs[H,T-1] > 75) then begin
         Eventcheck (EvSched,T,Eventmarker,1);
         if not Eventmarker then begin
            XM := Remhrs[H,T-1] - 75;
            KM := ScheduleC (XM);
            while (C1Sched[(T-1)*4+1+KM].WorkC >= 2) and (KM <= 2)
             do begin
               KM := KM + 1;
            end;
            KB := 1;
            while (C1Sched[(T-1)*4+1+KM].CSchedarray[KB].Occup)
              and (KB <= 5) do begin
               KB := KB + 1;
               if KB >= 5 then begin
                  writeln (Outputfile,
' Schedule conflict C1 inspection for helicopter ',H,' in month ',T);
               end;
            end;
            C1Sched[(T-1)*4+1+KM].CSchedarray[KB].Occup := True;
            C1Sched[(T-1)*4+1+KM].CSchedarray[KB].Idnr := H;
            C1Sched[(T-1)*4+1+KM].WorkC :=
              C1Sched[(T-1)*4+1+KM].WorkC + 1;
         end;
      end;


      if (Remhrs[H,T] <= 150) and (Remhrs[H,T-1] > 150) then begin
         Eventcheck (EvSched,T,Eventmarker,2);
         if not Eventmarker then begin
            XM := Remhrs[H,T-1] - 150;
            KM := ScheduleC (XM);
            while (C2Sched[(T-1)*4+1+KM].WorkC >= 2) and (KM <= 2)
             do begin
               KM := KM + 1;
            end;
            KB := 1;
            while (C2Sched[(T-1)*4+1+KM].CSchedarray[KB].Occup)
              and (KB <= 5) do begin
               KB := KB + 1;
               if KB >= 5 then begin
                  writeln (Outputfile,
   ' Schedule conflict C2 inspection for helicopter ',H,' in month ',T);
               end;
            end;
            C2Sched[(T-1)*4+1+KM].CSchedarray[KB].Occup := True;
            C2Sched[(T-1)*4+1+KM].CSchedarray[KB].Idnr := H;
```

```
                C2Sched[(T-1)*4+1+KM].WorkC :=
                  C2Sched[(T-1)*4+1+KM].WorkC + 1;
             end;
          end;

          if (Remhrs[H,T] <= 225) and (Remhrs[H,T-1] > 225) then begin
             Eventcheck (EvSched,T,Eventmarker,3);
             if not Eventmarker then begin
                XM := Remhrs[H,T-1] - 225;
                KM := ScheduleC (XM);
                while (C1Sched[(T-1)*4+1+KM].WorkC >= 2) and
                  (KM <= 2) do begin
                   KM := KM + 1;
                end;
                KB := 1;
                while (C1Sched[(T-1)*4+1+KM].CSchedarray[KB].Occup)
                  and (KB <= 5) do begin
                   KB := KB + 1;
                   if KB >= 5 then begin
                      writeln (Outputfile,
    ' Schedule connflict C1 inspection for helicopter ',H,' in month ',T);
                   end;
                end;
                C1Sched[(T-1)*4+1+KM].CSchedarray[KB].Occup := True;
                C1Sched[(T-1)*4+1+KM].CSchedarray[KB].Idnr := H;
                C1Sched[(T-1)*4+1+KM].WorkC :=
                   C1Sched[(T-1)*4+1+KM].WorkC + 1;
             end;
          end;
       end;
    end;
    Printinputstruc (Input,Outputfile,Remhrs);
    Printschedule (EvSched,DSched,C1Sched,C2Sched,Outputfile);
    Printstatistic (Outputfile,Remhrs,Y);
    Printhours (Outputfile,X,SumY);
    Graphicschedule (DSched,C1Sched,C2Sched,EvSched,Outputfile);
    close (Inputfile);
    close (Outputfile);
    close (Eventfile);
 end.
```

B. RESULTING OUTPUT

Data Set 1991, solved with solver OSL and solution procedure Cascade; listed partly only;

Helicopterdata
Nr = order of helicopters w.r.t. remaining hours to next D inspection;
ID = helicopter identification;
Remhrs = remaining hours to next inspection (both levels);
Nextinsp = code numbers 1..16 for next inspection;
init.Remhrs = remaining hours to next D inspection;

Nr	ID	Remhrs	Nextinsp	init.Remhrs
1	43	0	12	0
2	44	0	16	0
3	45	0	4	0
4	42	2	8	2
5	41	5	4	5
6	40	7	16	7
7	1	10	4	10

8	2	12	8	12
9	3	13	12	13
10	4	15	16	15
11	5	20	4	20
12	6	35	8	35
13	7	45	12	45
14	8	60	16	60
15	9	75	4	75
16	10	5	7	80
17	11	15	11	90
18	12	20	15	95
19	13	35	3	110
20	14	60	7	135
21	15	70	11	145
22	16	75	15	150
23	17	5	2	155
24	18	15	6	165
25	19	25	10	175
26	20	35	14	185
27	39	45	10	195
28	38	50	6	200
29	37	55	2	205
30	36	60	14	210
31	35	70	10	220
32	34	75	6	225
33	33	5	1	230
34	32	10	13	235
35	31	15	9	240
36	30	17	5	242
37	29	25	1	250
38	28	35	13	260
39	27	45	9	270
40	26	47	5	272
41	25	53	1	278
42	24	65	13	290
43	23	75	9	300
44	22	75	5	300
45	21	75	1	300

****
* **Schedule** *   Remark: week 5 and weeks 10 - 48 omitted;
****

Code for Inspections during an event :
1 = C1 prior to a C2
2 = C2
3 = C1 after a C2

****

```
Week =  1
D-Schedule
ID =   43   with D-Inspection : 3
ID =   44   with D-Inspection : 4
ID =   45   with D-Inspection : 1

C1-Schedule
ID =   33

C2-Schedule

Event-Schedule

*********************************************
Week =  2
D-Schedule
ID =   43   with D-Inspection : 3
ID =   44   with D-Inspection : 4
ID =   45   with D-Inspection : 1

C1-Schedule
ID =   11
ID =   12

C2-Schedule

Event-Schedule

*********************************************
Week =  3
D-Schedule
ID =   43   with D-Inspection : 3
ID =   44   with D-Inspection : 4
ID =   45   with D-Inspection : 1

C1-Schedule
ID =   31
ID =   29

C2-Schedule
ID =   19

Event-Schedule

*********************************************
Week =  4
D-Schedule
ID =   44   with D-Inspection : 4
ID =   40   with D-Inspection : 4
ID =    5   with D-Inspection : 1
```

```
C1-Schedule

C2-Schedule

Event-Schedule

*********************************************
Week =   10
D-Schedule
ID =    8   with D-Inspection : 4

C1-Schedule
ID =   32
ID =   21

C2-Schedule
ID =   33

Event-Schedule
Eventnumber = 1  ID =   20  Inspection = 1
Eventnumber = 1  ID =   22  Inspection = 3
Eventnumber = 1  ID =   24  Inspection = 3
Eventnumber = 1  ID =   25  Inspection = 2
Eventnumber = 1  ID =   27  Inspection = 3
Eventnumber = 1  ID =   30  Inspection = 3
Eventnumber = 1  ID =   31  Inspection = 2
Eventnumber = 1  ID =   34  Inspection = 2
Eventnumber = 1  ID =   35  Inspection = 2
Eventnumber = 1  ID =   43  Inspection = 3

*********************************************
```

Percentage of operationally ready Helicopters

```
80.0 %  at Month   1
84.4 %  at Month   2
84.4 %  at Month   3
86.7 %  at Month   4
88.9 %  at Month   5
80.0 %  at Month   6
82.2 %  at Month   7
82.2 %  at Month   8
84.4 %  at Month   9
80.0 %  at Month  10
84.4 %  at Month  11
86.7 %  at Month  12
```

Planned Hours per month and helicopter

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	10	0	0	0	0	0	30	30	0	0	30
2	12	0	0	0	0	30	30	30	62	30	0	0
3	0	0	0	0	0	13	0	0	0	30	30	30
4	15	0	0	0	0	0	0	0	0	0	0	65
5	20	0	30	0	0	30	30	0	0	30	30	30
6	0	30	5	30	30	30	30	30	62	0	0	0
7	0	6	0	0	30	9	0	30	62	30	30	0
8	0	30	30	30	30	30	30	30	0	30	0	0
9	15	0	0	0	30	30	0	0	62	30	0	30
10	0	0	30	0	30	0	0	20	0	30	30	0
11	30	0	30	30	0	0	0	0	0	0	30	30
12	30	30	0	0	30	5	0	30	62	0	30	30
13	0	30	0	30	0	0	0	0	30	20	0	65
14	0	0	0	30	30	30	0	30	0	15	30	30
15	30	30	0	30	30	25	0	0	0	30	0	0
16	0	0	30	30	75	15	0	0	30	0	0	65
17	0	30	0	30	30	30	0	0	0	30	5	0
18	0	30	30	30	30	30	0	0	0	15	27	0
19	30	30	30	30	0	0	0	0	0	0	0	11
20	30	30	100	0	25	0	0	0	30	0	0	30
21	30	30	30	0	0	0	30	0	62	0	0	0
22	0	0	100	0	0	30	30	30	0	0	0	0
23	30	30	0	0	0	30	30	30	62	30	0	0
24	0	0	100	30	19	30	0	0	0	0	30	30
25	30	30	100	0	0	30	0	0	0	0	0	65
26	0	0	30	30	75	30	30	0	30	30	0	17
27	0	0	100	0	30	0	30	0	30	0	30	0
28	0	0	0	30	0	30	30	0	30	30	0	30
29	30	30	0	0	30	30	30	0	62	30	0	0
30	0	0	100	0	30	0	0	0	0	30	30	0
31	30	30	100	30	30	0	0	20	30	0	0	0
32	0	0	30	0	30	30	0	30	0	30	30	30
33	30	30	30	30	75	30	5	30	0	0	0	0
34	30	30	100	0	0	0	0	30	30	5	0	0
35	0	0	100	30	0	30	30	30	0	0	0	0
36	0	0	0	2	30	0	30	0	62	30	30	0
37	30	0	0	30	30	30	0	0	30	30	25	0
38	30	30	0	0	75	30	0	5	30	0	0	0
39	30	0	0	30	75	30	11	19	0	0	0	30
40	7	0	0	30	75	30	0	30	0	30	0	65
41	5	0	0	0	0	30	30	30	62	0	30	0
42	2	0	0	0	75	30	0	30	30	30	0	0
43	0	0	100	0	0	30	30	30	30	30	30	0
44	0	0	0	30	75	30	30	30	13	30	30	0
45	0	0	0	0	30	0	30	0	30	0	30	30

Number of D - inspections per month

3 inspections in month 1
2 inspections in month 2
2 inspections in month 3
2 inspections in month 4
2 inspections in month 5
2 inspections in month 6
4 inspections in month 7
2 inspections in month 8
2 inspections in month 9
4 inspections in month 10
2 inspections in month 11
2 inspections in month 12

Graphical Schedule

1 = D1, 2 = D2, 3 = D1a, 4 = D3, * = C1, $ = C2, E = Event
Week = week equivalent (48 for a year) on horizontal axis
Helicopter Identification on vertical axis

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	2	2	2	2	-	-
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	1	1	1	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-	2	2	2	2	-	-
7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	4	4	4	4	4	-	-
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	*	-	-	-	-	-	-	-
11	-	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	*	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	*	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	$	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	$	-	-	-	-	-	-	-	-	*	-
19	-	-	$	-	-	-	-	-	-	-	-	-	-	*	-	-
20	-	-	-	-	$	-	-	-	E	E	E	E	-	-	-	-
21	-	-	-	-	-	-	-	-	-	*	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	E	E	E	E	-	-	-	-
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	E	E	E	E	-	-	-	-
25	-	-	-	-	-	-	*	-	E	E	E	E	-	-	-	-
26	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*	-
27	-	-	-	-	-	-	-	-	E	E	E	E	-	-	-	-
28	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
29	-	-	*	-	-	-	-	-	-	-	-	-	-	-	-	-
30	-	-	-	-	-	-	-	-	E	E	E	E	-	-	-	-
31	-	-	*	-	-	-	-	-	E	E	E	E	*	-	-	-
32	-	-	-	-	-	-	-	-	-	*	-	-	-	-	-	-
33	*	-	-	-	-	-	-	-	-	$	-	-	-	-	-	-
34	-	-	-	-	-	-	-	-	E	E	E	E	-	-	-	-
35	-	-	-	-	-	-	-	-	E	E	E	E	-	-	-	-
36	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
37	-	-	-	-	-	-	-	-	-	-	-	-	-	-	$	-
38	-	-	-	-	-	$	-	-	-	-	-	-	-	-	-	-
39	-	-	-	-	-	-	-	-	-	-	-	-	-	$	-	-
40	-	-	-	4	4	4	4	4	-	-	-	-	-	-	-	-
41	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	1
42	-	-	-	-	-	-	-	-	-	-	2	2	2	2	-	-
43	3	3	3	-	-	-	-	-	E	E	E	E	-	-	-	-
44	4	4	4	4	4	-	-	-	-	-	-	-	-	-	-	-
45	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-

	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	-	-	-	-	-	-	1	1	1	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	-	-	-	*	-	-
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	3
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	*	-	-	-	-	-	-
6	-	-	-	-	-	*	-	-	-	-	-	-	-	-	$	-
7	-	-	-	-	-	-	-	3	3	3	-	-	-	-	-	-
8	-	-	-	-	-	*	-	-	-	-	-	-	-	-	$	-
9	-	-	-	-	-	-	1	1	1	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	2
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	4	4	4	4	4	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	*	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	3	3	3	-	-	-	-
16	-	E	E	-	-	-	-	-	-	-	4	4	4	4	4	-
17	-	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	4	4	4	4	4	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-	$	-	-	-	-	-	-
23	-	-	-	-	-	-	*	-	-	-	-	-	-	-	-	$
24	-	$	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26	-	E	E	-	-	-	-	-	-	-	-	-	-	-	-	-
27	-	$	-	-	-	-	-	-	-	-	-	-	-	-	-	-
28	-	-	-	-	*	-	-	-	-	-	-	-	-	-	-	-
29	-	-	-	-	-	$	-	-	-	-	-	-	-	-	-	-
30	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
31	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3
32	-	-	-	-	-	-	$	-	-	-	-	-	-	-	-	-
33	-	E	E	-	-	-	-	-	-	-	-	1	1	1	-	-
34	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
35	-	-	-	-	-	-	*	-	-	-	-	-	-	-	-	-
36	-	-	-	-	-	-	-	-	-	-	$	-	-	-	-	-
37	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
38	-	E	E	-	-	-	-	-	-	-	-	-	-	-	-	-
39	-	E	E	-	-	-	-	-	-	-	-	-	-	-	-	-
40	-	E	E	-	-	-	-	-	-	-	-	-	-	$	-	-
41	1	-	-	-	-	-	-	-	-	-	-	-	-	*	-	-
42	-	E	E	-	-	-	-	-	-	-	-	-	-	-	-	-
43	-	-	-	-	-	-	-	-	-	-	-	-	-	$	-	-
44	-	E	E	-	-	-	-	-	-	$	-	-	-	-	-	-
45	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
1	-	-	-	-	-	-	-	-	-	-	-	-	-	*	-	-
2	-	-	E	-	-	-	-	-	-	-	-	-	-	-	-	-
3	3	-	-	-	-	-	-	-	-	-	-	-	-	*	-	-
4	-	-	-	-	-	-	4	4	4	4	4	-	E	E	E	E
5	-	-	-	-	-	-	-	-	-	-	$	-	-	-	-	-
6	-	-	E	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	E	-	-	-	-	-	-	-	$	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	-	-	E	-	-	*	-	-	-	-	-	-	-	-	-	-
10	2	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	3	3	3	-	-	-	-	-	-	-	-	-	-	-
12	-	-	E	-	-	-	-	-	-	-	-	-	-	-	$	-
13	-	-	-	-	-	-	1	1	1	-	-	-	E	E	E	E
14	-	-	-	-	-	-	2	2	2	2	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	E	E	E	E
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	2	2	2	2	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	E	-	-	-	-	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23	-	-	E	-	-	-	-	*	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	-	-	-	-	*	-	-	-
25	-	-	-	-	-	-	-	-	-	-	-	-	E	E	E	E
26	*	-	-	-	-	-	-	-	-	-	-	-	-	-	2	2
27	-	-	-	-	-	-	-	-	*	-	-	-	-	-	-	-
28	-	$	-	-	-	-	-	-	-	-	-	-	-	-	-	-
29	-	-	E	-	-	-	-	-	-	-	-	-	-	-	-	-
30	-	-	-	-	-	-	-	-	*	-	-	-	-	-	-	-
31	3	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
32	-	-	-	-	-	-	-	-	-	*	-	-	-	-	-	-
33	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
34	-	-	-	-	-	-	-	-	-	-	2	2	2	2	-	-
35	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
36	-	-	E	-	-	-	*	-	-	-	-	-	-	-	-	-
37	-	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-
38	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
39	-	-	3	3	3	-	-	-	-	-	-	-	-	-	-	-
40	-	-	-	-	-	-	-	-	-	-	-	-	E	E	E	E
41	-	-	E	-	-	-	-	-	-	-	-	-	-	-	-	-
42	-	$	-	-	-	-	-	-	-	-	-	-	-	-	-	-
43	-	-	-	-	-	-	*	-	-	-	-	-	-	-	-	-
44	-	-	-	-	-	*	-	-	-	-	-	-	-	-	-	-
45	-	*	-	-	-	-	-	-	-	-	-	-	-	-	$	-

APPENDIX D IMPLEMENTATION OF SHORT TERM PLANNING SYSTEM

```
    $TITLE   MISSION ASSIGNMENT MODEL
    $STITLE  A.SGASLIK  Thesis

* Selection of helicopters for specified missions

*---------CONTROL OPTIONS-------------------------------------------

$OFFUPPER OFFSYMLIST OFFSYMXREF

OPTIONS
    LIMCOL = 0       , LIMROW = 0      , SOLPRINT = OFF    , DECIMALS = 2
    RESLIM = 100000 , ITERLIM = 5000, OPTCR    = 0.05   ,  SEED     = 3141
    OPTION LP = XA, RMIP = XA, MIP = XA;

*---------DEFINITIONS AND DATA---------------------------------------

SETS
   M  mission identification  /1*20/
   H  helicopter identification /1*45/
   I  simultaneous mission groups /1*6/ ;

ALIAS (H,HA);

PARAMETERS
   REMINSP(H)    remaining hours to next inspection
         / 1     0
           2     0
           3     5
           4     5
           5     5
           6     10
           7     10
           8     10
           9     12
           10    15
           11    20
           12    25
           13    25
           14    30
           15    35
           16    35
           17    35
           18    40
           19    45
           20    45
           21    50
           22    50
           23    55
           24    55
           25    60
           26    60
           27    70
           28    75
           29    75
           30    74
           31    60
           32    60
           33    40
           34    40
           35    30
           36    30
           37    30
           38    20
           39    10
           40    5
           41    5
           42    5
           43    0
           44    0
           45    0 /

   AVAIL(H)      availability of helicopter for mission
         / 1     0
           2     0
           3     1
```

```
4     1
5     1
6     0
7     1
8     1
9     1
10    1
11    0
12    1
13    0
14    1
15    1
16    1
17    1
18    1
19    0
20    1
21    1
22    1
23    1
24    1
25    1
26    1
27    1
28    1
29    1
30    1
31    1
32    1
33    1
34    1
35    0
36    1
37    0
38    1
39    1
40    1
41    1
42    1
43    0
44    0
45    0 /

  OPERATION(H)  operation limitation
* code description  0=none, 1=CTP only, . VFR only, 3= daylight only
  / 1     0
    2     0
    3     0
    4     0
    5     0
    6     1
    7     0
    8     2
    9     0
    10    0
    11    0
    12    0
    13    0
    14    0
    15    3
    16    3
    17    0
    18    0
    19    0
    20    0
    21    0
    22    0
    23    0
    24    0
    25    0
    26    3
    27    0
    28    0
    29    1
    30    2
    31    2
    32    0
    33    0
    34    0
    35    0
    36    0
    37    1
    38    1
    39    1
```

```
          40    0
          41    0
          42    0
          43    0
          44    0
          45    0 /

 INSPWEEK(H)    week of next inspection out of yearly schedule
        / 1     12
          2     12
          3     13
          4     13
          5     14
          6     14
          7     15
          8     15
          9     16
          10    17
          11    17
          12    17
          13    18
          14    19
          15    20
          16    21
          17    22
          18    23
          19    24
          20    25
          21    26
          22    26
          23    26
          24    27
          25    28
          26    29
          27    30
          28    31
          29    32
          30    33
          31    33
          32    30
          33    29
          34    29
          35    28
          36    27
          37    26
          38    25
          39    19
          40    15
          41    15
          42    15
          43    12
          44    13
          45    12 /

  PRIORITY(H)   user given priority for helicopter
* 0 = lowest  3 = highest
        / 1     0
          2     0
          3     3
          4     3
          5     3
          6     2
          7     2
          8     2
          9     2
          10    2
          11    1
          12    1
          13    1
          14    1
          15    1
          16    3
          17    3
          18    3
          19    3
          20    2
          21    2
          22    2
          23    1
          24    0
          25    0
          26    0
          27    1
```

```
             28    1
             29    1
             30    1
             31    3
             32    2
             33    3
             34    2
             35    1
             36    1
             37    0
             38    0
             39    1
             40    2
             41    3
             42    3
             43    0
             44    0
             45    0 /

      EQUIPMENT(H) equipment code for each helicopter
* 1=11 seats  2=5seats  3=400kg  5=ITL  6=ITR  7=Winch
           / 1     1
             2     2
             3     3
             4     4
             5     5
             6     6
             7     1
             8     2
             9     3
             10    4
             11    5
             12    6
             13    1
             14    2
             15    3
             16    4
             17    5
             18    6
             19    1
             20    2
             21    3
             22    4
             23    5
             24    6
             25    1
             26    2
             27    2
             28    2
             29    2
             30    3
             31    3
             32    3
             33    3
             34    3
             35    3
             36    2
             37    2
             38    2
             39    1
             40    3
             41    2
             42    5
             43    6
             44    2
             45    3 /

      NEXTINSP(H)   completion length of next inspection in weeks
           / 1      0.5
             2      1
             3      3
             4      4
             5      5
             6      0.5
             7      0.5
             8      0.5
             9      0.5
             10     1
             11     1
             12     3
             13     0.5
             14     0.5
             15     1
             16     0.5
             17     3
```

```
          18    4
          19    4
          20    5
          21    0.5
          22    1
          23    0.5
          24    1
          25    3
          26    4
          27    0.5
          28    0.5
          29    0.5
          30    1
          31    1
          32    5
          33    0.5
          34    1
          35    3
          36    4
          37    5
          38    0.5
          39    4
          40    3
          41    1
          42    0.5
          43    0.5
          44    0.5
          45    1  /

   LENGTH(M)     length of mission in hours
         / 1    2
           2    2
           3    2
           4    4
           5    5
           6    5
           7    5
           8    5
           9    5
           10   1
           11   4
           12   20
           13   5
           14   10
           15   6
           16   7
           17   8
           18   5
           19   3
           20   5 /

   REQUEQU(M)    required equipment code for mission
 * 1=11 seats  2=5seats  3=400kg  5=ITL  6=ITR  7=Winch
         / 1    2
           2    2
           3    2
           4    3
           5    4
           6    5
           7    5
           8    3
           9    2
           10   1
           11   2
           12   6
           13   6
           14   1
           15   1
           16   1
           17   1
           18   5
           19   3
           20   2 /

   SPAREREQU(M)  spare helicopter required for mission
         / 1    0
           2    0
           3    0
           4    1
           5    1
           6    1
           7    1
           8    0
```

```
          9   0
          10  0
          11  1
          12  1
          13  1
          14  1
          15  1
          16  0
          17  0
          18  0
          19  0
          20  1 /

   MISSIONLIM(M) not acceptable helicopter operation limitation
* 1=CTP only   2=VFR only   3=Daylight only   4=none
         / 1   4
           2   4
           3   4
           4   1
           5   1
           6   1
           7   1
           8   2
           9   2
           10  2
           11  3
           12  3
           13  3
           14  3
           15  1
           16  4
           17  4
           18  4
           19  4
           20  4 /


   MISGROUP(M)  groups missions which are happening simultaneously
         / 1   1
           2   1
           3   2
           4   2
           5   2
           6   2
           7   3
           8   3
           9   3
           10  3
           11  3
           12  4
           13  4
           14  4
           15  4
           16  5
           17  5
           18  6
           19  6
           20  6 /;


SCALARS
   CONST1     constant for objective function  / 3 /
   CONST2     constant for objective function  / 5 /;

*---------MODEL-----------------------------------------------------

BINARY VARIABLE
   FM(H,M)    one if helicopter is selected for mission

   FS(H,M)    one if helicopter is selected as spare for mission ;


LOOP (M $ (SPAREREQU(M) EQ 0), FS.FX(H,M) = 0 );


POSITIVE VARIABLES
   PENMULT(H)    penalty for assigning one helo for more then one mission ;

VARIABLE
   COST     objective function variable;
```

```
EQUATIONS
   OBJ          objective function equation
   CHO(M)       exactly one helicopter for each mission
   CHOS(M)      exactly one spare for each mission when required
   HRS(H)       length of mission constraint
   MULT(H)      multiple missions constraint
   MULTS(H,I)   multiple missions constraint for simultaneous missions;


OBJ..    COST =E= 100 +
         SUM((H,M) $ (AVAIL(H)), ((CONST1 - PRIORITY(H)) +
                              0.1 * (CONST2 - NEXTINSP(H)) +
         MAX( (EQUIPMENT(H) - REQUEQU(M)) , (REQUEQU(M) - EQUIPMENT(H)) ) +
                              0.1 * INSPWEEK(H) ) * FM(H,M) )  +
         SUM((H,M) $ (AVAIL(H) AND SPAREREQU(M)),
         ( 0.5 * (CONST1 - PRIORITY(H)) + 0.1 * (CONST2 - NEXTINSP(H)) +
         MAX( (EQUIPMENT(H) - REQUEQU(M)) , (REQUEQU(M) - EQUIPMENT(H)) ) +
                              0.1 * INSPWEEK(H) ) * FS(H,M) )  +
         SUM(H, 0.5 * PENMULT(H) );


CHO(M)..     SUM(H $ (AVAIL(H) AND (OPERATION(H) NE MISSIONLIM(M))),
             FM(H,M) ) =E= 1;


CHOS(M) $ (SPAREREQU(M))..     SUM(H $ (AVAIL(H) AND (OPERATION(H)
                               NE MISSIONLIM(M))), FS(H,M) ) =E= 1;

HRS(H) $ (AVAIL(H))..          REMINSP(H) =G=
                               SUM(M, LENGTH(M)*(FM(H,M) + FS(H,M)));

MULTS(H,I) $ (AVAIL(H))..      SUM(M $ (MISGROUP(M) EQ ORD(I)), FM(H,M) +
                               FS(H,M) ) =L= 1 ;

MULT(H) $ (AVAIL(H))..         SUM(M, FM(H,M) ) =L= 1 + PENMULT(H);


MODEL MISSION  /ALL/;
SOLVE MISSION  USING  MIP MINIMIZING COST;

* intermediate non-integer solution :
If (MISSION.MODELSTAT EQ 9,  MISSION.ITERLIM = 10000;
                               MISSION.OPTCR = 0.1;
                               SOLVE MISSION  USING  MIP MINIMIZING COST );

* infeasible solution or still intermediate non - integer solution :
If ((MISSION.MODELSTAT EQ 9) OR (MISSION.MODELSTAT EQ 4) OR
    (MISSION.MODELSTAT EQ 10),
                               LOOP (H, PRIORITY(H) = 3);
                               MISSION.OPTCR = 0.2;
                               SOLVE MISSION  USING  MIP MINIMIZING COST );


*---------REPORTS-----------------------------------------------------

DISPLAY  FM.L;
DISPLAY  FS.L;
DISPLAY  PENMULT.L;
```

APPENDIX E INTERFACE PROPOSALS

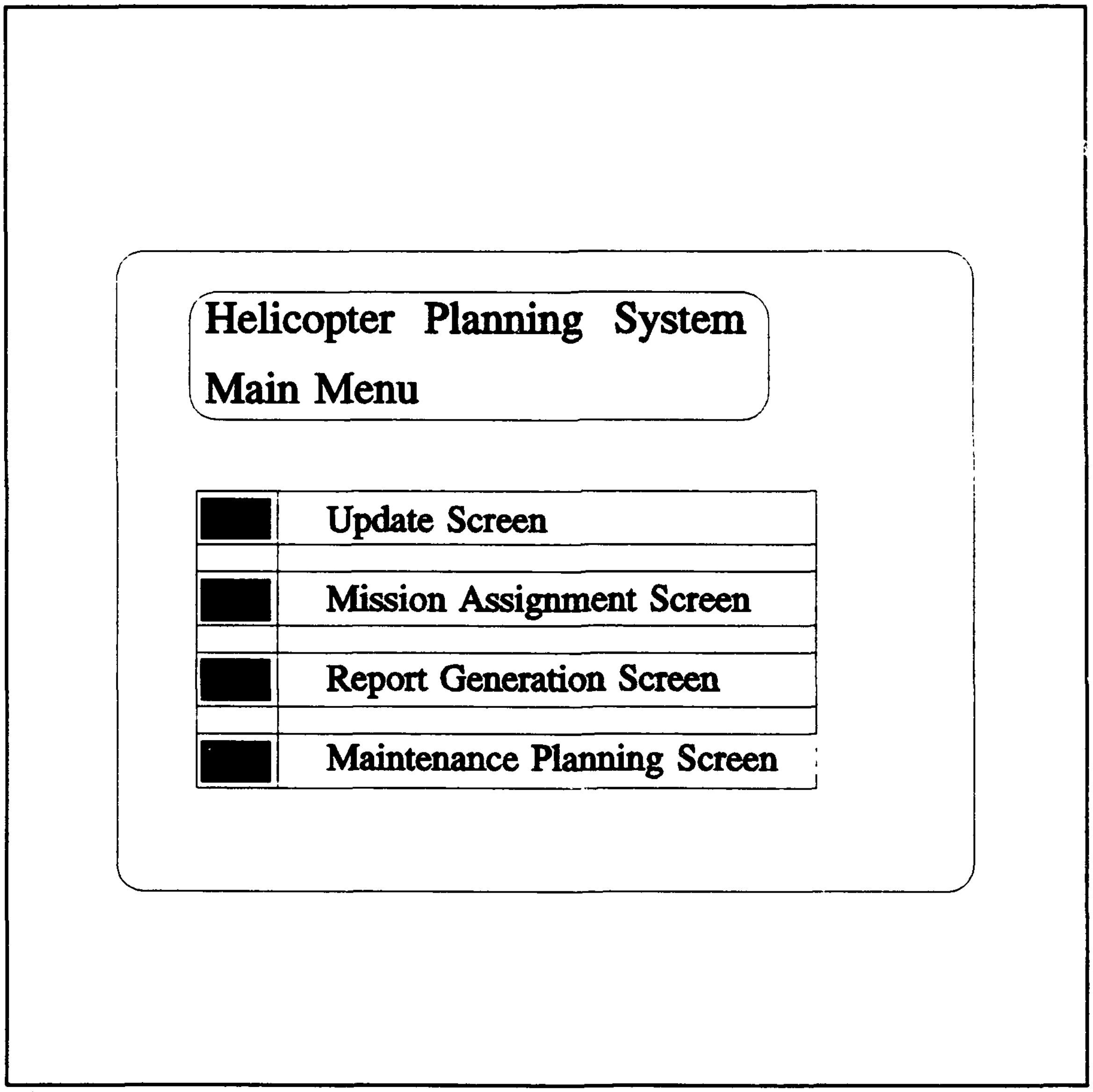

Figure 4

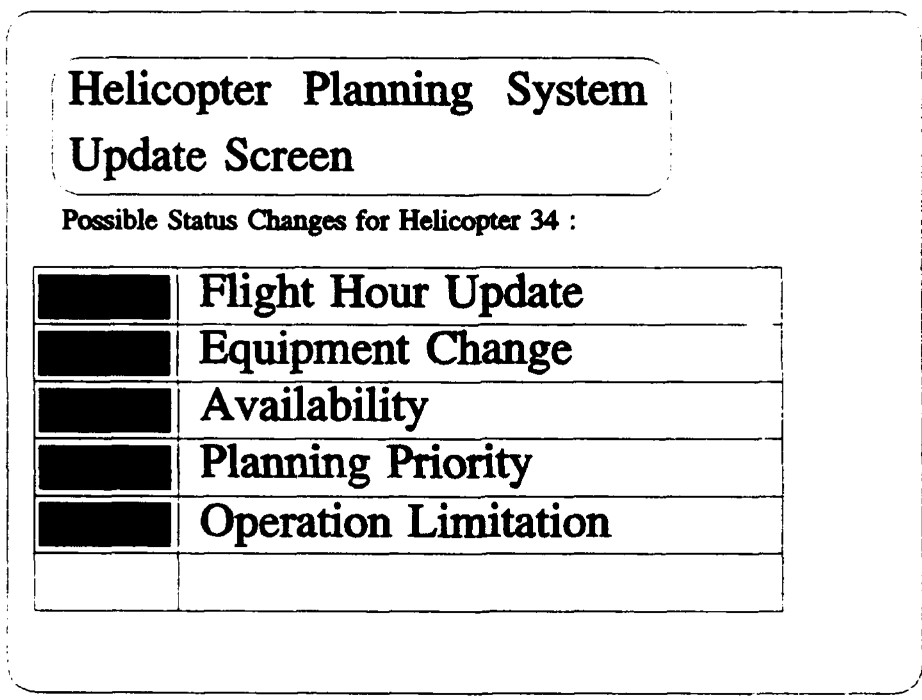
Helicopter Planning System
Update Screen
Possible Status Changes for Helicopter 34 :
Flight Hour Update
Equipment Change
Availability
Planning Priority
Operation Limitation

Figure 5

Helicopter Planning System
Flight Hour Update

Helicopter 34

Current Flight Hours : 635 hrs 45 min;

Next Inspection C1 in 35 hrs 45 min;

_ _ _ hours _ _ minutes at _ _ / _ _ / 93

Figure 6

Helicopter Planning System
Equipment Change

Helicopter 34 Current Equipment Status : 5 Seats ;

	5 Seats
	11 Seats
	400 kg Load
	Internal Tank Left
	Internal Tank Right
	Winch

Figure 7

Helicopter Planning System
Availability Code

Helicopter 34 Current Availability : Operational

C1	Failure Fuselage
C2	Failure Engine
D1	Failure Avionic
D2	Failure Electric
D1a	Mission
D3	Operational

Estimated Completion Time : _ _ / _ _ / 93

Figure 8

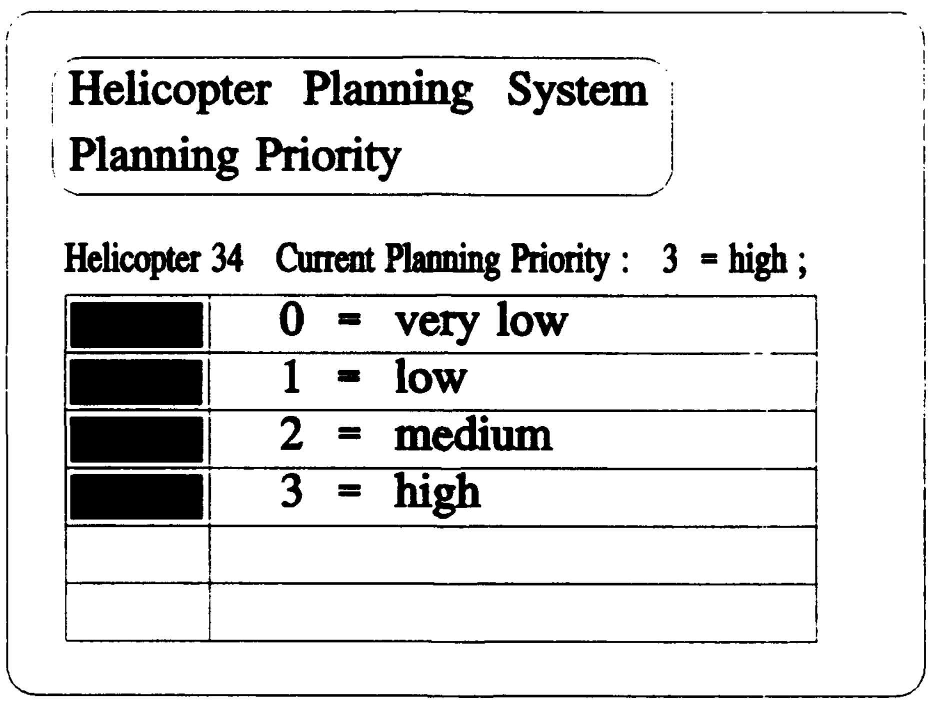
Helicopter Planning System
Planning Priority
Helicopter 34 Current Planning Priority : 3 = high ;
0 = very low
1 = low
2 = medium
3 = high

Figure 9

Helicopter Planning System
Operation Limitation

Helicopter 34 Current Limitation : 0 = none ;

	0 = none
	1 = CTP only
	2 = VFR only
	3 = daylight only

Figure 10

Helicopter Planning System
Mission Assignment Screen

Next Page | Update | Optim

Page 1 11/11/93 - 12/11/93

Nr	Grp	Hrs	Start	End	Description	Equ	Lim	Helo	Equ	Prio	Fix
1	1	5	0700	1500	CTP	5	0	45	5	3	
2	1	5	0700	1500	CTP	5	0	42	4	2	
3	1	4	0700	1200	CTP	1	2	5	1	3	
4	1	4	0900	1600	BFE 345	6	2	18	5	3	
					spare	6	2	23	4	2	
5	2	5	1700	2330	CTP	2	2	23	4	2	

Legend: Lim = not allowed operation limitation code
Equ = Equipment code Prio = Priority code

Figure 11

Helicopter Planning System
Maintenance Planning

Schedule Week : 21 - 22

Yearly Schedule

Helo	Insp	Week
2	C1	22
12	C2	21
23	C1	22
42	D1	21
45	C1	22

<= 10 hours left

Helo	Insp	Hours
12	C2	9
14	C1	7
23	C1	2
42	D1	5
45	C1	9

Short Term Schedule

Helo	Insp	Week
12	C2	22
14	C1	22
23	C1	21
42	D1	21
45	C1	22

Print

Figure 12

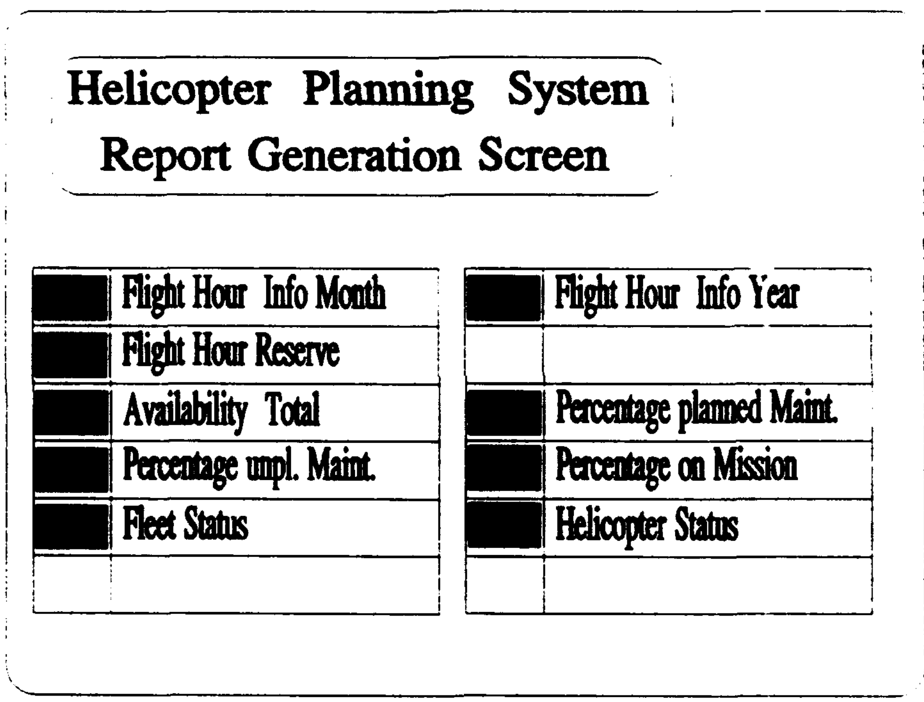
Helicopter Planning System
Report Generation Screen
Flight Hour Info Month
Flight Hour Reserve
Availability Total
Percentage unpl. Maint.
Fleet Status
Flight Hour Info Year
Percentage planned Maint.
Percentage on Mission
Helicopter Status

Figure 13

LIST OF REFERENCES

Borland Turbo Pascal, Version 6.0, User's Guide, Borland International, Inc., Scotts Valley, California, 1990.

Brooke, A.,Kendrick, D., and Meeraus, A., *GAMS: A User's Guide*, Second Edition, The Scientific Press, South San Francisco, California, 1992.

Brown, Gerald G., Clemence, Robert D., Teufert, William R., and Wood, R. Kevin, *An Optimization Model for Modernizing the Army's Helicopter Fleet*, Interfaces 21, pp.39-52, July-August 1991.

Christensen, Dennis K., and Pasadilla, Magno, O., *The Design of a Personal Computer Database for the Expert System Advisor for Aircraft Maintenance Scheduling (ESAAMS)*, Master's Thesis, Naval Postgraduate School, Monterey, California, December 1991.

Hackett, Stephen B., and Pennartz, Sam E., *Decision Support Systems: An Approach to Aircraft Maintenance Scheduling in the Strategic Air Command*, Master's Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio School of Systems and Logistic, September 1982.

OSL: Optimization Subroutine Library, Guide and Reference, IBM Corporation, Kingston, New York, 1991.

Shenolikar, A., *Automating the decision support for ATE operations management*, AUTOTESTCON, Proceedings of the Conference, Fort Worth, Texas, November 1-3, Institute of Electrical and Electronics Engineers, Inc., New York, 1983.

Talluri, K., and Gopalan, R., *An Eulerian Tour Problem in Aircraft Maintenance Routing*, Paper presented at the ORSA/TIMS Joint National Meeting, Phoenix, Arizona, November 1993.

Wallace, David E., *Analysis Tools for United States Air Force Sortie Optimization and Munitions Planning*, Master's Thesis, Naval Postgraduate School, Monterey, California, December 1992.

XA: Professional Linear Programming System, GAMS Version 2.2, Sunset Software Technology, San Marino, California, 1993.

INITIAL DISTRIBUTION LIST

No.Copies

1. Defense Technical Information Center 2
   Cameron Station
   Alexandria, Virginia 22304-6145

2. Library, Code 052 2
   Naval Postgraduate School
   Monterey, California 93943-5002

3. Professor Robert F. Dell, Code OR/De 2
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California 93943-5002

4. Professor Gordon H. Bradley, Code OR/Bz 1
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California 93943-5002

5. Cpt Sgaslik Achim 5
   Am Madenbach 4
   97708 Bad Bocklet
   Germany